BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

PROFESSIONAL PROGRAMMING COURSES

TABLE OF CONTENTS

ADO.NET COURSE ..... *pg 1*

What is ADO.NET? ..... *pg 2*

Advantages of ADO.NET ..... *pg 2*

New Features of ADO.NET ..... *pg 3*

ADO.NET Course Outline ..... *pg 3*

ANDROID PROGRAMMING COURSE ..... *pg 4*

What is Android? ..... *pg 5*

What is Android Programming? ..... *pg 5*

Advantages of Android Programming Language ..... *pg 5*

Features of Android Programming Language ..... *pg 6*

Android Programming Course Outline ..... *pg 7*

ARDUINO PROGRAMMING COURSE ..... *pg 8*

What is Arduino? ..... *pg 9*

History of Arduino ..... *pg 10*

Features of Arduino ..... *pg 11*

Benefits and advantages of Arduino ..... *pg 11*

Parts of an Arduino Board ..... *pg 12*

The Arduino Family ..... *pg 12*

ASSEMBLY LANGUAGE COURSE ..... *pg 13*

What is Assembly Language? ..... *pg 14*

Features of Assembly Language ..... *pg 15*

Benefits of Assembly Language ..... *pg 16*

Why Study Assembly Language? ..... *pg 16*

COBOL COURSE ..... *pg 17*

What is COBOL? ..... *pg 18*

Evolution of COBOL ..... *pg 19*

Features of COBOL ..... *pg 19*

Benefits Of COBOL ..... *pg 19*

Character Set in COBOL ..... *pg 20*

COMPUTER PROGRAMMING COURSE ..... *pg 21*

What is Computer Programming? ..... *pg 22*

Advantages of Computer Programming ..... *pg 22*

Computer Programming Tasks ..... *pg 23*

Computer Programming Features ..... *pg 24*

Types of Computer Programming Languages ..... *pg 25*

Computer Programming Course Outline ..... *pg 26*

TABLE OF CONTENTS

C++ PROGRAMMING COURSE ..... *pg 27*

What is C++ Programming? ..... *pg 28*

Features of C++ Programming ..... *pg 29*

Benefits and Advantages of C++ Programming ..... *pg 30*

C++ Programming Study ..... *pg 31*

C++ Programming Course Outline ..... *pg 32*

C PROGRAMMING COURSE ..... *pg 33*

What is C Programming? ..... *pg 34*

Features of C Programming ..... *pg 35*

Benefits of C Programming ..... *pg 36*

Relationship To Other Languages ..... *pg 37*

Various Versions of C ..... *pg 37*

Syntax ..... *pg 37*

Character Set ..... *pg 37*

Reserved Words ..... *pg 38*

Operators ..... *pg 38*

C-SHARP PROGRAMMING COURSE ..... *pg 39*

What is C# or C-Sharp? ..... *pg 40*

Features Of C# Programming ..... *pg 41*

Why Study C# Programming? ..... *pg 42*

Benefits Of C# Programming Language ..... *pg 43*

C# Course Outline ..... *pg 44*

D PROGRAMMING LANGUAGE COURSE ..... *pg 45*

What is D Programming? ..... *pg 46*

Features Of D Programming ..... *pg 47*

Benefits And Advantages Of D Programming ..... *pg 48*

Development Tools for D Programming ..... *pg 48*

Example of D Language Syntax ..... *pg 49*

Why You Should Learn D Programming ..... *pg 49*

D Programming Course Outline ..... *pg 50*

DATA STRUCTURE AND ALGORITHMS COURSE ..... *pg 51*

What is Data Structure & Algorithms ..... *pg 52*

Fundamentals Of Data Structure & Algorithms ..... *pg 53*

Advantages of Data Structure & Algorithms ..... *pg 54*

Data Structure and Algorithms Course Outline ..... *pg 55*

TABLE OF CONTENTS

FLUTTER COURSE ..... ***pg 56***

What is Flutter? ..... *pg 57*

How Does Flutter Work? ..... *pg 58*

Benefits Of Flutter ..... *pg 59*

Compatibility ..... *pg 59*

Rapid development is possible due to the use of hot reloading ..... *pg 59*

It is cost-effective and time-saving ..... *pg 60*

Excellent performance ..... *pg 60*

Open-source ..... *pg 60*

Flutter Course Outline ..... *pg 61*

FORTRAN COURSE ..... *pg 62*

What is Fortran? ..... *pg 63*

Why Learn Fortran? ..... *pg 63*

Fortran Basics ..... *pg 64*

Advantages of Fortran ..... *pg 67*

Main Features of Fortran ..... *pg 67*

Fortran Course Outline ..... *pg 68*

F# PROGRAMMING LANGAUGE COURSE ..... *pg 69*

What is F# or F-Sharp? ..... *pg 70*

F# Development Tools ..... *pg 70*

Why Study F#? ..... *pg 71*

Benefits of F# Programming Language ..... *pg 72*

F# Programming Course Outline ..... *pg 73*

GO PROGRAMMING LANGUAGE COURSE ..... *pg 74*

What is Go Programming Language? ..... *pg 75*

Features of Go Programming Language ..... *pg 75*

Applications of Go Programming Language ..... *pg 76*

Advantages of Go Programming Language ..... *pg 77*

Go Programming Language Benefits ..... *pg 77*

Go Programming Language Course Outline ..... *pg 78*

IONIC COURSE ..... *pg 79*

What is Ionic? ..... *pg 80*

Features Of Ionic ..... *pg 81*

Benefits Of Ionic ..... *pg 82*

Ionic Course Outline ..... *pg 83*

TABLE OF CONTENTS

PROGRAMMABLE LOGIC CONTROLLERS (PLC) COURSE ..... *pg 84*

What is Programmable Logic Controllers (PLC)? ..... *pg 85*

Features of Programmable Logic Controllers? ..... *pg 86*

Benefits of Programmable Logic Controllers ..... *pg 87*

Programmable Logic Controllers Course Outline ..... *pg 88*

IOS COURSE ..... *pg 89*

What is iOS? ..... *pg 90*

Features of IOS ..... *pg 91*

Benefits of IOS ..... *pg 91*

Career Opportunities in IOS ..... *pg 92*

IOS Course Outline ..... *pg 93*

IPHONE BASICS COURSE ..... *pg 94*

What is iPhone? ..... *pg 95*

Features of an iPhone ..... *pg 96*

Benefits of an iPhone ..... *pg 97*

Why Study iPhone Basics ..... *pg 98*

iPhone Basics Course Outline ..... *pg 99*

IPAD BASICS COURSE ..... *pg 100*

What is an iPad? ..... *pg 101*

Features of an iPad ..... *pg 101*

Benefits of an iPad ..... *pg 102*

Why Study iPad Basics ..... *pg 102*

iPad Basics Course Outline ..... *pg 103*

JAVA PROGRAMMING COURSE ..... *pg 104*

What is JAVA? ..... *pg 105*

Features Of JAVA Programming ..... *pg 106*

Benefits Of JAVA Programming ..... *pg 107*

JAVA Programming Course Outline ..... *pg 108*

JULIA PROGRAMMING LANGUAGE COURSE ..... *pg 109*

What is Julia Programming? ..... *pg 110*

Features and Benefits of Julia Programming Language ..... *pg 111*

Why Study Julia Programming Language ..... *pg 112*

Julia Programming Course Outline ..... *pg 113*

TABLE OF CONTENTS

K PROGRAMMING COURSE ..... ***pg 114***

What is K Programming? ..... ***pg 115***

Features of K Programming ..... ***pg 116***

Benefits of K Programming ..... ***pg 117***

Why Study K Programming ..... ***pg 117***

K Programming Course Outline ..... ***pg 118***

MATLAB COURSE ..... ***pg 119***

What Is MATLAB? ..... ***pg 120***

Features Of MATLAB ..... ***pg 121***

Benefits Of MATLAB ..... ***pg 122***

MATLAB Study ..... ***pg 122***

NODE JS COURSE ..... ***pg 123***

What Is Node JS? ..... ***pg 124***

Features Of Node JS ..... ***pg 124***

Why Study Node JS? ..... ***pg 125***

Career Opportunities in Node JS ..... ***pg 126***

Benefits And Advantages Of Node JS ..... ***pg 126***

Node JS Course Outline ..... ***pg 127***

PASCAL PROGRAMMING LANGUAGE ..... ***pg 128***

What is Pascal? ..... ***pg 129***

Features of Pascal ..... ***pg 130***

Benefits of Pascal ..... ***pg 130***

Facts About Pascal ..... ***pg 130***

Compilers and Interpreters for Pascal ..... ***pg 131***

PYTHON PROGRAMMING COURSE ..... ***pg 132***

What is Python? ..... ***pg 133***

Features Of Python Programming ..... ***pg 134***

Benefits Of The Python Programming Language ..... ***pg 135***

Career Opportunities in Python Programming ..... ***pg 135***

Python Course Outline ..... ***pg 136***

REACT NATIVE COURSE ..... ***pg 137***

What is React Native? ..... ***pg 138***

Features and Benefits of React Native ..... ***pg 139***

Why Study React Native? ..... ***pg 140***

React Native Course Outline ..... ***pg 141***

TABLE OF CONTENTS

R Programming Course ..... ***pg 142***

What is R Programming? ..... ***pg 143***

Features of R Programming ..... ***pg 144***

Benefits of R Programming Language ..... ***pg 144***

Applications of R Programming ..... ***pg 145***

Rexx Programming Course ..... ***pg 146***

What is Rexx Programming? ..... ***pg 147***

Features of Rexx Programming ..... ***pg 147***

Benefits of Studying Rexx Programming ..... ***pg 148***

REXX Programming Course Outline ..... ***pg 149***

Rust Programming Course ..... ***pg 150***

What is Rust Programming Language? ..... ***pg 151***

Features Of Rust Programming Language ..... ***pg 152***

Benefits Of The Rust Programming Language ..... ***pg 152***

Why Study Rust Programming Language? ..... ***pg 153***

Rust Programming Language Course Outline ..... ***pg 153***

Scala Course ..... ***pg 154***

What is Scala Programming? ..... ***pg 155***

Features of Scala Programming ..... ***pg 156***

Benefits of Scala Programming ..... ***pg 157***

Why Study Scala Programming? ..... ***pg 157***

Scala Web Frameworks ..... ***pg 157***

Concepts in Scala Programming ..... ***pg 158***

Basic Syntax ..... ***pg 158***

Scala Programming Course Outline ..... ***pg 159***

Swift Programming Course ..... ***pg 160***

What is Swift Programming? ..... ***pg 161***

Features of Swift Programming ..... ***pg 162***

Benefits of Swift Programming ..... ***pg 162***

Swift Programming Course Outline ..... ***pg 163***

Visual Basic.Net Course ..... ***pg 164***

What Is Visual Basic.NET? ..... ***pg 165***

Features Of Visual Basic.NET ..... ***pg 166***

Benefits Of VB.NET ..... ***pg 166***

TABLE OF CONTENTS

VISUAL STUDIO.NET COURSE ..... *pg 167*

What is VisualStudio .NET? ..... *pg 168*

Features of Visual Studio.Net ..... *pg 169*

Benefits of VisualStudio.Net ..... *pg 170*

Tools in VisualStudio.Net ..... *pg 170*

Supported Products ..... *pg 170*

Why Study Visual Studio.Net? ..... *pg 171*

Visual Studio.Net Course Outline ..... *pg 171*

VISUAL BASIC COURSE ..... *pg 172*

What is Visual Basic? ..... *pg 173*

Features Of Visual Basic ..... *pg 174*

Benefits Of Visual Basic ..... *pg 175*

Visual Basic Course Outline ..... *pg 176*

XAMARIN COURSE ..... *pg 177*

What is Xamarin? ..... *pg 178*

Features Of Xamarin ..... *pg 179*

Benefits Of Xamarin ..... *pg 180*

Xamarin Course Outline ..... *pg 180*

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

ADO.NET COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is ADO.NET?

ActiveX Data Objects (ADO.NET) is an application program interface developed by Microsoft which allows programmers to have access to a relational or non-relational database from both Microsoft and other database providers. ADO has a set of Component Object Model (COM) objects for accessing data sources.

ADO is a part of MDAC (Microsoft Data Access Components), which provides a middleware layer between programming languages and OLE DB. ADO allows a developer to write programs that can access data without knowing how the database is implemented or constructed; except developers must be aware of the database for connection only.

No knowledge of SQL is required to access a database when using ADO, even though you can use ADO to execute SQL commands directly. For instance, if you wanted to write a program that would provide users of your Web site with data from an IBM DB2 database or an Oracle database, you could include ADO program statements in an HTML file that you then saved as an Active Server Page. Then, when a user requested the page from the Web site, the page sent back would contain exact data from a database, obtained using ADO code.

Advantages of ADO.NET

1. ADO.NET offers Single Object-Oriented API
2. It offers Interoperability
3. It Increases Productivity
4. XML Support
5. Managed Code
6. Visual Data Components
7. Performance and Scalability
8. Job Opportunity

New Features of ADO.NET

1. Bulk Copy Operation
2. Batch Update
3. Data Paging
4. Connection Details
5. DataSet.RemotingFormat Property
6. DataTable's Load and Save Methods
7. New Data Controls
8. DbProvidersFactories Class
9. Customized Data Provider
10. DataReader's New Execute Methods

In the Full Course, you will learn everything you need to know about **ADO.NET** with **Certification** to showcase knowledge/skill gained.

ADO.NET Course Outline

ADO.NET Introduction
ADO.NET Data Providers
ADO.NET SQL Server
ADO.NET Connection
ADO.NET Command
ADO.NET DataReader
ADO.NET DataSet
ADO.NET DataAdapter
ADO.NET DataTables
ADO.NET WF Example
ADO.NET MVC Example
ADO.NET Video Lectures
ADO.NET Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

ANDROID PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Android?

Android is simply an open source and Linux-based operating system for mobile devices like smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other top tech companies.

What is Android Programming?

Android programming is the method by which applications are built and developed for the Android devices operating system. Applications are often built in Java programming language using the Android software development kit (SDK), however other development environments and programming languages are also available.

Android Programming is programmed or written in the Java programming language and use Java core libraries. Its applications are composed of one or more applications components which are : activities, services, content providers, and broadcast receivers.

Advantages of Android Programming Language

1. Android is an open source programmig language, so it is robust with very large developer community which keeps on interacting with each other and also gives information about the upcoming versions of Android.

2. Easy to integrate and customize.

3. It enhances security.

4. It provides scope for innovation and it has a global presence.

5. It has free software development tools like IDE Android development studio for the developer community which reduces the development costs for the development company.

6. It provides lots of job opportunities with promising salary.

7. It provides self-employment opportunity.

Features of Android Programming Language

The Android software development kit (SDK) has an entire set of development tools. These tools include a debugger, libraries, a simple handset emulator based on QEMU, documentation, sample code, and tutorials. Some currently supported development platforms are computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows 7 or later.

Enhancements to Android's SDK are interconnected with the overall Android platform development. The SDK also have supports for older versions of the Android platform in case developers wish to target their applications at older devices. The development tools are also downloadable components, so after downloading the latest version and platform, you can also download older platforms and tools for compatibility testing.

Android applications are packaged and compiled in .apk format which is stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package also contains .dex files (compiled byte code files known as Dalvik executables), resource files, etc. Android SDK package contains some toolkit like Android Debug Bridge. This consists of both client and server-side programs which corresponds with one another. The ADB is usually accessed through the command-line interface, even if many other graphical user interfaces exist to control ADB.

Android is a popular computing platform use on Linux operating system. The first commercial version of Android was introduced to the market in the year 2008 in the form of a mobile phone platform, back when the most popular cell phones for a business user was the Blackberry, when the iPhone was beginning to make meaningful waves across all sectors and when the majority of phone users were still tapping out texts from a flip phone.

The Android is a layered environment, one that is built upon a foundation of the Linux kernel and includes rich functionality. The users interface subsystem includes everything you would expect from a mature operating system environment including windows, views, and widgets for displaying common elements like edit boxes, lists or drop-down lists. The browser is both capable for general web browsing and available for embedding directly into your own applications.

Android is a powerful operating system supporting a large number of applications in smartphones. These applications make life more comfortable and advanced for the users and they are as followed : activities, services content providers, and broadcast receivers etc. Each component performs a different role in the overall application behavior and each one can be activated individually.

Android is a powerful operating system competing with Apple 4Gs and supports great features, such features are Beautiful UI, Connectivity, Storage, Media support, Multi touch, Multi tasking, Multi language, Wi-Fi direct, Android beam, Messaging and so on.

Android Applications Development, to begin with, requires the following software, which is: Java JDK5 or Later version and Android Studio. There are so many sophisticated technologies that are available to develop Android applications, the common technologies, which are predominantly used are: Android Studio and Eclipse IDE. You can download and install the software to start development.

In the Full Course you will learn everything you need to know about **Android Programming Language** with **Certificate** to showcase your knowledge and competence.

Android Programming Course Outline

1. Android Programming Basics
Android Programming - Introduction
Android Programming - Environment Setup
Android Programming - Architecture
Android Programming - Application Components
Android Programming - Hello World Example
Android Programming - Resources
Android Programming - Activities
Android Programming - Services
Android Programming - Broadcast Receivers
Android Programming - Content Providers
Android Programming - Fragments
Android Programming - Intents/Filters
2. Android Programming User Interface
Android Programming - UI Layouts
Android Programming - UI Controls
Android Programming - Event Handling
Android Programming - Styles and Themes
Android Programming - Custom Components
3. Android Programming Advanced Concepts
Android Programming - Drag and Drop
Android Programming - Notifications
Android Programming - Location Based Services
Android Programming - Sending Email
Android Programming - Sending SMS
Android Programming - Phone Calls
Publishing Android Application
4. Android Programming Useful Examples
Android Programming - Alert Dialoges
Android Programming - Animations
Android Programming - Audio Capture
Android Programming - AudioManager
Android Programming - Auto Complete
Android Programming - Best Practices
Android Programming - Bluetooth
Android Programming - Camera
Android Programming - Clipboard
Android Programming - Custom Fonts
Android Programming - Data Backup
Android Programming - Developer Tools
Android Programming - Emulator
Android Programming - Facebook Integration
Android Programming - Gestures
Android Programming - Google Maps
Android Programming - Image Effects
Android Programming - ImageSwitcher
Android Programming - Internal Storage
Android Programming - JetPlayer
Android Programming - JSON Parser
Android Programming - LinkedIn Integration
Android Programming - Loading Spinner
Android Programming - Localization
Android Programming - Login Screen
Android Programming - MediaPlayer
Android Programming - Multitouch
Android Programming - Navigation
Android Programming - Network Connection
Android Programming - NFC Guide
Android Programming - PHP/MySQL
Android Programming - Progress Circle
Android Programming - ProgressBar
Android Programming - Push Notification
Android Programming - RenderScript
Android Programming - RSS Reader
Android Programming - Screen Cast
Android Programming - SDK Manager
Android Programming - Sensors
Android Programming - Session Management
Android Programming - Shared Preferences
Android Programming - SIP Protocol
Android Programming - Spelling Checker
Android Programming - SQLite Database
Android Programming - Support Library
Android Programming - Testing
Android Programming - Text to Speech
Android Programming - TextureView
Android Programming - Twitter Integration
Android Programming - UI Design
Android Programming - UI Patterns
Android Programming - UI Testing
Android Programming - WebView Layout
Android Programming - Wi-Fi
Android Programming - Widgets
Android Programming - XML Parsers
5. Android Programming Video Lectures
6. Android Programming Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

ARDUINO PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Arduino?

Arduino is an open-source prototype platform, based on easy to find and use hardware and software.

Arduino is made up of a circuit board, which can be programmed (generally referred to as a microcontroller) and a ready-made software known as Arduino IDE, that is used to write and upload the computer software program to the physical board.

Arduino products are licensed under the GNU Lesser General Public License (LGPL) or under the GNU General Public License (GPL), which permits anyone to manufacture and distribute Arduino software and boards. Arduino boards are made commercially available in the preassembled form or as a do-it-yourself (DIY) kits.

Arduino board designs make use of a variety of controllers and microprocessors. The boards are fully equipped with assortments of digital and analog input/output (I/O) pins that may be directly interfaced with various other expansion boards ('shields') or breadboards (used for prototyping) and other circuits.

The boards highlight various serial communications interfaces, some of which include Universal Serial Bus (USB) on some models, that are also used for loading software programs from personal computers. The microcontrollers can be programmed by using languages such as C and C++. In addition to using conventional compiler toolchains, the Arduino project presents an Integrated Development Environment (IDE) that is based on the Processing language project.

It is aimed at providing a low-cost and easy way for professionals and novices to create devices that interact with their environment by means of and actuators. Typical examples of such devices that are intended for beginner hobbyists include thermostats, simple robots, and motion detectors. The name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project usually meet up.

The Arduino project kickstarted at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. during that time the scholars were using a BASIC Stamp microcontroller at a price of $50, a considerable expense for various students the year 2003 Hernando Barragán created the event platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas. Casey Reas is understood for co-creating, with Ben Fry, the Processing development platform. The project goal was to make simple, low-cost tools for creating digital projects by non-engineers. But rather than continuing the work on Wiring, they forked the project and renamed it, Arduino.

The founding Arduino core team was made up of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis. Following the completion of the Wiring platform, lighter and fewer expensive versions were distributed within the open-source community.

Federico Musto, Arduino's former CEO, got half ownership of the corporation. In the month of April 2017, Wired said that Federico Musto had forged his academic record on his company's website, and even on some Italian business docs, Musto was not until recently posted as holding a Ph.D. from the Massachusetts Institute of Technology. But in some other cases, his biography also claimed an MBA from NY University. Wired said that none of the universities had a record of Musto's attendance, and Musto later agreed in an interview that he doesn't have any of those degrees. Around that very same time, Massimo Banzi announced that the Arduino Foundation would be a fresh start for Arduino but a year later, the inspiration still hasn't been established, and therefore the state of the project remains unclear.

Features of Arduino

Below are some of the features of Arduino

1. Arduino boards are designed to be able to read either analog or digital input signals from different sensors and then convert it into an output such as turning LED on/off, activating a motor, connect to the cloud and many other actions.

2. You can manage the functions of your board by sending a series of instructions to the microcontroller on the board through the Arduino IDE (known as uploading software).

3. Unlike most preceding programmable circuit boards, Arduino does not require an extra piece of hardware in order to load new code onto the board. You can simply use a USB cable to transfer the code.

4. Additionally, the Arduino IDE makes use of a simplified version of C++, therefore making it easier to learn and program.

5. Finally, Arduino presents you with a standard form factor that breaks down the functions of the micro-controller into a more accessible package.

Benefits and advantages of Arduino

Below are some of the benefits of Arduino

1. Arduino kits are always cheap.

2. It comes with an open supply of hardware features that allows its users to develop their own kit.

3. The software of the Arduino is fully-fitted with all kinds of operating systems like Windows, Linux, and Macintosh, etc.

4. It also comes with an open supply software system trait that allows tough software system developers to use the Arduino code to blend with the current programing language libraries and may be extended and changed.

5. For beginners, Arduino is very simple to use.

6. You are exposed to different job opportunities when you learn how to use Arduino.

7. You can command high paying hardware jobs when you learn Arduino.

Parts of an Arduino Board

1. Power (USB/Barrel Jack).
2. Pins (3,3V, 5V, Analog, GND, AREF, Digital, PWM).
3. Rest Button.
4. Power LED Indicator.
5. TX RX LEDs.
6. Main IC.
7. Voltage Regulator.

The Arduino Family

1. Arduino Uno (R3): The Arduino Uno is actually a really great choice for your very first Arduino board. It has everything you need to kickstart your Arduino journey. It contains every single thing you need to work with the microcontroller, easily connect it to a PC with a USB cable or turn it on with an AC-to-DC adaptor or power pack or battery to get started.
2. LilyPad Arduino.
3. RedBoard.
4. Arduino Mega.
5. Arduino Leonardo.
6. Sensors.
7. Shields.

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

ASSEMBLY LANGUAGE COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Assembly Language?

Assembly Language is a low-level computer programming language that is used for microprocessors and other programmable devices. Assembly is not just a single programming language, instead, it is a group of languages.

Assembly Language makes use of a symbolic representation of the machine code that is needed to program a particular CPU architecture. Assembly languages are also known as Assembly codes. The term is frequently used together with 2GL.

Assembly Language is the most fundamental programming language that is available for any computer processor. With Assembly Language, a programmer would work with only the operations that are directly implemented on the physical CPU. Assembly Languages basically lack high-level features such as variables, loops, and functions, and they are not transferable between the various families of computer processors. They have the same structures, features, and groups of commands as a machine language, but they allow a programmer to make use of names instead of just binary numbers. This language is still very useful for computer programmers when speed is a necessary factor or when they need to carry out a method that is not possible in high-level programming languages.

Assembly Language is developed to understand syntax instruction and convert it directly to machine language for further processing. Assembly language depends on the system's architecture, whether it is the operating system or computer system's architecture.

Assembly Language is mainly made up of mnemonic processor instructions or data that is for the processor, and other statements or instructions. It is developed with the help of compiling the high-level language source code like C, C++, Java, C#, etc. Assembly Language helps in fine-tuning the software program.

The study of Assembly Language is still very important for computer programmers. It helps in taking absolute control over the system and in the allocation of its resources. By learning assembly language, the programmer would be able to write the code to gain access registers and be able to retrieve the memory address of pointers and values. It mainly helps in speeding up the optimization process that increases the efficiency and performance of software systems.

Features of Assembly Language

There are many features of Assembly Language and some of them are:
1. Instruction Set: An instruction set architecture is an abstract model of a computer. It is also referred to as architecture or computer architecture.

2. Bundling: A bundle contains up to three instructions and an associated template. The template defines which type of execution unit processes each instruction in the bundle.

3. Instruction Groups: Instructions that are allowed to execute in parallel are organized in instruction groups. An instruction group is a set of consecutive instructions that should have no interdependencies. The instruction group is terminated by a stop (;;).

4. Data Allocation: IAS enables allocating and initializing space in memory. IAS supports these data types.

5. Assembly Language Directives: IAS supports all Itanium architecture assembly language directives except local label directives. The supported directives include the following operations or information: section control, symbol control, file inclusion, bundle template selection, debug information, unwind information.

6. 64-bit Address Space: 64-bit describes a generation of computers in which 64-bit processors are the norm. 64 bits is a word size that defines certain classes of computer architecture, buses, memory, and CPUs and, by extension, the software that runs on them. IAS supports 64-bit address space.

7. Alignment: By default, IAS aligns bundles on 16-byte boundaries, and data elements according to their size. IAS aligns each section according to the largest alignment request in the section. Bundles, data elements, or a .align directive create alignment requests

8. Assignment Statements: Assignment statements enable the programmer to define a symbol by assigning it a value. This value may be a reference to another symbol, register name, or expression.

9. Aliasing: This is an effect that causes different signals to become indistinguishable (or aliases of one another) when sampled. IAS supports the aliasing of symbol names and section names.

10. Arithmetic Expression Handling: IAS supports the use of arithmetic expressions for constants and addresses, using standard arithmetic notation. Arithmetic expressions can include symbols, numeric constants, and operators.

Benefits of Assembly Language

There are many benefits of Assembly Language and some of them are:

1. Assembly Language allows the programmer to carry out more complex jobs to run in a simpler way.

2. Assembly codes are memory-efficient since they require lesser memory space to operate.

3. Programs written in Assembly languages are faster in speed, as its time of execution is lesser.

4. Assembly Languages programs are mainly hardware-oriented i.e they are written with just the hardware in mind.

5. It requires less instruction to get the result.

6. It is used for critical jobs.

7. With Assembly Language, you are not required to keep track of memory locations.

8. It is a low-level embedded system, hence it can be used to program any embedded system.

Why Study Assembly Language?

1. Gain complete control over a system's resources.

2. Understand processor and memory function.

3. Direct access to hardware.

4. Career opportunities and advancement.

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

COBOL COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is COBOL?

COBOL which is the acronym for Common Business-Oriented Language is an English-like compiled computer programming language that was designed for use in organizations and businesses. It is procedural, imperative and an object-oriented software development language. COBOL was basically used in finance, business, and administrative systems for organizations, companies, and governments.

COBOL is still used widely in legacy software that is deployed on mainframe computers, such as for large-scale batch and transaction processing projects. But due to the downward and declining popularity together with the retirement of experienced COBOL developers, COBOL programs are being migrated to more recent platforms, rewritten in recent, high-level languages or simply replaced with software packages. Most development in COBOL is now primarily just to manage and maintain already existing applications.

COBOL was developed in 1959 by the team at CODASYL and it was initially and partially based on the preceding computer programming language design work by Grace Hopper, regularly referred to as "the (grand)mother of COBOL". COBOL was developed as part of an effort of the US Department of Defense to create a portable computer programming language for data analysis and processing. It was primarily seen as a stopgap, but the Department of Defense instantaneously, forced computer manufacturing firms to provide it, accounting for its widespread adoption.

COBOL was standardized in 1968 and has since been revised 4 times. Expansions to the language include support for both structured and object-oriented programming concepts. The current standard of COBOL is ISO/IEC 1989:2014.

COBOL statements are written in an English-like syntax, which was developed to be both self-documenting and highly readable. It is verbose and has an array of over 300 reserved words. In contrast with recent, succinct syntax, COBOL has a more English-like syntax.

COBOL code is divided into four divisions which are: environment, identification, data, and procedure which contains a rigid hierarchy of sections, sentences, and paragraphs. Lacking in the areas of a large standard library, the current standard specifies 43 statements, 87 functions, and just one class.

Academic computer scientists were generally not interested in business applications when COBOL was developed and was not involved in its design. It was designed effectively from the ground up as a computer language for business, with a huge emphasis on the inputs and outputs, whose only data types are just numbers and strings of text.

COBOL has been underrated throughout its life, for its verbose, design process, and poor support for structured programming concepts. These weaknesses brought about a monolithic and, though designed to be English-like, not easily understandable and verbose programs.

Evolution of COBOL

In the era of the 1950s, when the businesses began increasing in the west side of the world, A need to automate various processes for ease of operation grew and this gave rise to a high-level programming language meant for business data processing.

1. In 1959, COBOL was developed by CODASYL (Conference on Data Systems Language).
2. The next version, COBOL-61, was released in 1961 with some revisions.
3. In 1968, COBOL was approved by ANSI as a standard language for commercial use (COBOL-68).
4. It was again revised in 1974 and 1985 to develop subsequent versions named COBOL-74 and COBOL-85 respectively.
5. In 2002, Object-Oriented COBOL was released, which could use encapsulated objects as a normal part of COBOL programming.

Features of COBOL

Below are some of the features of COBOL.

1. COBOL has a clearly defined syntax.
2. COBOL has a Metalanguage.
3. COBOL has a nice and highly structured code format.
4. Identification division
5. COBOL has support for object-oriented programming concepts.
6. COBOL has an environment division.

Benefits Of COBOL

Why you should consider COBOL.

1. COBOL is self-documented and easy to understand/read.
2. COBOL is able to handle large volumes of data easily.
3. COBOL is easy to use for the development and maintenance of business applications, that are portable across platforms without recompilation.
4. COBOL programs are robust.
5. COBOL can provide various employment opportunities for people that know it.
6. COBOL was the first widely used high-level programming language. It has an English-like language that is very user-friendly. All of the program can be coded in simple English words.
7. COBOL is used as a self-documenting language.
8. COBOL can handle huge data processing.
9. COBOL is compatible with its previous versions.
10. COBOL has effective error messages so much that the resolution of bugs is easier.

Character Set in COBOL

Characters are the lowest in the family tree and they cannot be further divided.
The COBOL Character Set consist 78 characters which are shown below –

Sr.No.	Character & Description	
1	A-Z	Alphabets(Upper Case)
2	a-z	Alphabets (Lower Case)
3	0-9	Numeric
4		Space
5	+	Plus Sign
6	-	Minus Sign or Hyphen
7	*	Asterisk
8	/	Forward Slash
9	$	Currency Sign
10	,	Comma
11	;	Semicolon
12	.	Decimal Point or Period
13	“	Quotation Marks
14	(	Left Parenthesis
15	)	Right Parenthesis
16	>	Greater than
17	<	Less than
18	:	Colon
19	‘	Apostrophe
20	=	Equal Sign

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

COMPUTER PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Computer Programming?

Computer Programming is the organized method and process of designing and building computer programs and applications to solve problems or to accomplish a specific computing task.

Programming have existed as far back as 1206 AD. Machine code was the early language of computer programs, as at then it was written in binary notations. Assembly language was later developed that allows programmer specify instruction in a text format.

The first standard computing program was introduced in 1843, when a mathematician named Ada Lovelace developed an algorithm to solve a sequence of Bernoulli numbers using Charles Babbage's Analytical Engine.

Advantages of Computer Programming

1. It enhances effective communication between the computer and the programmer.

2. It extend the work of a computer to function according to programmer's algorithm.

3. It is used to build softwares and applications used in various fields and sectors such medicals, agriculture, accounting, management etc.

4. It is used to build both desktop, web and mobile applications.

5. It serves as source of self-employment.

6. It provides lots of job opportunities globally.

7. It provides applications that makes life easier.

8. It is widely used for complex mathematical calculations

9. It enhances business operations and growth.

10. Computer programs have increased our socialization via the use of web and mobile applications for chats, business e.t.c

11. Computer programs help us simplify our work, save much time and increase productivity at work places.

Computer Programming Tasks

Computer programming tasks generally involves these major steps:

1. Analysis of Problem and Solution,
2. Generating Algorithms for the Solution,
3. Profiling Algorithms' Accuracy,
4. Ascertaining Resource Consumption,
5. Choosing the right programming language for the job.
6. The Implementation of Algorithms in a chosen or preferred programming language known as coding.

Computer Programming Features

Some of the common features of any computer programming language:

1. Programmer or Coder: The person who codes or writes programming language algorithm to solve problems or to achieve a specifc computing task.

2. Algorithms: can be defined as the step by step approach or method in solving a given problem. Algorithm expression involves the use of variables and data types.

3. Variables: are memory location for storing data in a computer program. The value of variables can change during the execution of the program. We have two types of variables:
- String Variables and
- Numeric Variables.

4. Data Types: are data attributes which instructs the compiler or interpreter how the programmer intends to use the data.
Data types that exist across major programming languages like C, Java and Python are:
- Ch,
- int,
- arrays,
- list,
- numbers (float, decimals etc),
- string etc.

5. Coding Tools: The tools used in writing programs ranges from Text Editors, IDEs, Compilers and Interpreters.
- Compilers: A compiler is used to convert the program file to binary file.
- Text Editors: These are softwares used in writing raw codes.
- Interpreters: Interpreters are used to read and execute programs directly without any conversion.
- IDEs: also known as Integrated Development Environment (IDE) is a software application that provides comprehensive coder interface environment to computer programmers for software development. An IDE usually consists of: source code editor, automation tools and a debugger.

Types of Computer Programming Languages

There are hundreds of Programming Languages in existence, ranging from low-level language to high level programming languages. The most popular and commonly used ones are as follows:

1. C Programming Language
2. C# Programming Language
3. C++ Programming Language
4. Fortran Programming Language
5. Pascal Programming Language
6. Perl Programming Language
7. PHP Programming Language
8. Python Programming Language
9. ASP Programming Language
10. Java Programming Language
11. Javascript Programming Language
12. LISP Language
13. Scheme Language etc.

In the Full course, you will learn everything you need to know about **Computer Programming** with **Certificate** to showcase your knowledge and competence.

Computer Programming Course Outline

Computer Programming - Introduction
Computer Programming - Basic Elements
Computer Programming - Environment
Computer Programming - Basic Syntax
Computer Programming - Data Types
Computer Programming - Variables
Computer Programming - Keywords
Computer Programming - Operators
Computer Programming - Decisions
Computer Programming - Loops
Computer Programming - Numbers
Computer Programming - Characters
Computer Programming - Arrays
Computer Programming - Strings
Computer Programming - Functions
Computer Programming - File I/O
Computer Programming - Summary
Computer Programming - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

C++ PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is C++ Programming?

C++ is an average-level Computer Programming Language that was developed by **Bjarne Stroustrup in 1979** at **Bell Labs a**s an improvement to the C Language and it was initially given the name "C with Classes" but later in 1983, it was given the name C++. C++ runs on a lot of platforms, such as Windows, Linux, Mac OS, and the various versions of UNIX.

C++ **Language** is a statically typed, compiled, multi-purpose, case-sensitive, Programming language that supports both Procedural, Object-Oriented, and Generic Programming. C++ is referred to as a middle-level language because it is made up of a combination of both high-level and low-level programming languages features. C++ is a superset of C, therefore, virtually any legal C program is a legal C++ program.

The ANSI standard in C++ is an attempt to make sure that C++ codes are portable meaning that a code you write for a Microsoft compiler will compile without errors, using a compiler on a Mac, Windows box, Linux machine or an Alpha.

C++ supports a lot of different programming styles. You can write in the style of C, Fortran Smalltalk, etc., You can also write in any language. Each style can achieve its aims effectively while managing runtime and space efficiently.

C++ currently ranks number 4 in the IEE spectrum Top Programming Language.

Features of C++ Programming

1. Simple: C++ is a simple language meaning that it comes with a structured approach to split the problem into smaller parts, it also comes with a rich set of library functions, data types, etc.

2. Machine Independent or Portable: C++ programs unlike assembly language, can be run on different machines with a little bit or no change. But it is not platform-independent.

3. Mid-level Programming Language: C++ is also used to carry out low-level programming functions. It is used to build system applications such as the kernel, system drivers, etc. It also supports the features of high-level languages. That is why it is being referred to as mid-level language.

4. Structured Programming Language: C++ is a structured Programming language meaning that we can divide the program into parts by defining and using functions. So, it is very easy to understand and modify.

5. Rich Library: C++ gives you a lot of inbuilt functions that enable the development process to be very fast.

6. Memory Management: C++ supports the feature of allocating memory dynamically. In C++ language, we can release the allocated memory at any time by calling the free() function.

7. Speed: The time of compilation and execution of C++ programs are very fast.

8. Pointer: C++ provides the feature of pointers. Programmers can directly communicate with the memory by using the pointers. You can use pointers for memory, program functions, data structures, array, etc.

9. Recursion: In C++, we can call a function that is inside another function. It ensures code reusability for every function.

10. Object-Oriented: C++ is an object-oriented programming language. OOPs makes software development and maintenance easier than just developing in Procedure-oriented programming language where it is not easy to manage and maintain if code grows as project size grows.

11. Compiler based: C++ is a compiler-based Programming language, this means that without compiling your code, your C++ Program cannot be executed. First, we need to compile our Program using the C++ compiler and then we can run the program.

Benefits and Advantages of C++ Programming

There are a lot of benefits and advantages of learning the C++ Programming Language and some of them are:

1. C++ is used by thousands of programmers in essentially every application domain.

2. It is highly recommended for writing device drivers and other software that rely directly on the manipulation of hardware under real-time constraints.

3. It can be used indirectly because the primary user interface of a PC running Windows is written in C++.

4. C++ supports a wide variety of programming styles and syntax.

5. C++ standard library gives a rich set of functions such as manipulating files, strings, etc.

6. C++ is Portable and Scalable

7. C++ is Object-oriented

8. C++ is Multi-paradigm

9. Low-level Manipulation

10. Memory Management

11. Large Community Support

12. Compatibility with the C Programming Language

13. C++ fully supports object-oriented programming, including the four pillars of object-oriented development such as:
Encapsulation,
Data hiding,
Inheritance,
Polymorphism.

14. Learning C++ is a wise investment because with skill in C++ you can command a high paying job and various employment opportunities due to its power and popularity.

C++ Programming Study

Here we highlight some of the areas we covered in this course, which includes:

Definition: C++ program can be defined as a collection of objects that communicate via invoking each other's methods.

The ANSI standard: The ANSI standard is an attempt to ensure that C++ is portable; that code you write for Microsoft's compiler will compile without errors, using a compiler on a Mac, UNIX, a Windows box, or an Alpha. The ANSI standard has been stable for a while, and all the major C++ compiler manufacturers support the ANSI standard.

The Semicolon: In C++, the semicolon is a statement terminator.

C++ Identifier: C++ Identifier is a name used to identify a variable, function, class, module, or any other user-defined item.

Trigraphs are a few characters with an alternative representation.

C++ Data Types is to store information on various data types like character, wide character, integer, floating-point, double floating-point, boolean, etc.

C++ Variable Types provides us with named storage that our programs can manipulate.

Variable Scope is a region of the program and they are the local and global variables.

Literals refer to fixed values that the program may not alter and they are integer, floating-point, Boolean, character and string literals.

Operators are symbols that tell the compiler to perform specific logical manipulations. They are the arithmetic operator, relational, logical, Bitwise, miscellaneous and assignment operators.

A Reference is another name for an already existing variable.

Data Structures are used to represent a record. To define a structure, you must use the struct statement. etc.

In The Full Course, you will learn everything you need to know about **C++ Programming Language** with **Certification** to showcase your knowledge and competence.

C++ Programming Course Outline

C++ Introduction
C++ Environment Setup
C++ Basic Syntax
C++ Comments
C++ Data Types
C++ Variable Types
C++ Variable Scope
C++ Constants/Literals
C++ Modifier Types
C++ Storage Classes
C++ Operators
C++ Loop Types
C++ Decision Making
C++ Functions
C++ Numbers
C++ Arrays
C++ Strings
C++ Pointers
C++ References
C++ Date & Time
C++ Basic Input/Output
C++ Data Structures
C++ Object Oriented
C++ Classes & Objects
C++ Inheritance
C++ Overloading
C++ Polymorphism
C++ Abstraction
C++ Encapsulation
C++ Interfaces
C++ Advanced
C++ Files and Streams
C++ Exception Handling
C++ Dynamic Memory
C++ Namespaces
C++ Templates
C++ Preprocessor
C++ Signal Handling
C++ Multithreading
C++ Web Programming
C++ Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

C PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is C Programming?

The C Programming Language is a multi-purpose, procedural software programming language that offers support for both structured programming, lexical variable scope, and recursion, while a static-type software system prevents developers from carrying out operations that are unintended. By design, the C Programming Language provides software developers with a language constructs that efficiently map to typical machine-level instructions and has found lasting use in applications that are previously coded in assembly language. Such applications include systems operating systems and various software applications for computers, from supercomputers to embedded systems.

The C Programming Language was originally developed at the Bell Laboratory by Dennis Ritchie in between 1972 and 1973 to make various utilities to run on Unix systems. Later, it was used for re-implementing the kernel of the Unix operating system. During the 1980s, the C programming language gradually gained popularity. It has become one of the most popular and widely used software programming languages, with lots of C compilers from various vendors available for the vast majority of existing computer and system architectures and operating systems. The C Programming Language has been standardized by the ANSI since 1989 by the International Organization for Standardization.

The C Programming Language is an imperative software procedural language. C was designed to be compiled by using a relatively straightforward and easy to use the compiler to provide low-level access to system memory and language constructs that would efficiently map the language syntax to it appropriate machine instructions, all with minimal support at runtime. Despite the low-level capabilities of C, the language was designed and developed to offer support for cross-platform programming. A standards-compliant C program source code that is developed with portability first in mind can be compiled for a vast majority of computer platforms and software operating systems with very few changes to its source code. The language is made available on various platforms, from simple embedded microcontrollers to large and complex supercomputers.

Features of C Programming

There are many features of the C Programming Language and some of them are:

1. C is a robust software programming language with a rich set of built-in features, functions, and operators that can be used to write any complex computer program.

2. The C compiler merges the abilities of a system assembly language with the features of a high-level programming language.

3. Programs that are written in C are very fast and efficient. This is due to the variety of data types and the powerful operators that are in C.

4. C programs are much faster than programs that are written in BASIC.

5. C programs are highly portable, this means that programs, once they are written can be run on multiple separate machines with little or no modification to the source code.

6. Another important feature of the C program is its ability to extend itself, c programs are scalable, modular and can be easily reused in other programs.

7. A program that is written in C is basically a collection of functions that are supported by the C library.

8. The C language is currently the most widely used software programming language in operating systems and embedded system development.

Benefits of C Programming

There are many benefits of the C Programming Language and some of them are:

1. The C programming language is the building block for many other programming languages.

2. C is a powerful and efficient language.

3.C programs are portable.

4. C comes with lots of built-in functions.

5. C has the quality to extend itself.

6. C is a structured programming language.

7. C is a middle-level language.

8. C allows for the implementation of algorithms and data structures.

9. C is a procedural programming language.

10. C has a dynamic memory allocation.

11. C is used for system-programming.

12. Learning C programming can get you high paying job opportunities because C programmers are very much sort after.

Relationship To Other Languages

A lot of new languages have gotten support directly or indirectly from C, alongside Limbo, Unix's C shell, C++, C#, D, Go, Java, JavaScript (including transpilers), LPC, Objective-C, Perl, PHP, Python, Rust, Swift, Verilog and SystemVerilog (hardware description languages) These languages have gotten many of their controlling structures and other base characteristics from C. Many of them (Python being an exception) also have highly identical syntax to C, and they combine the identical expression and statement syntax of C with underlying type systems, data models, and semantics that can be radically different.

Various Versions of C

1. K&R C
2. ANSI C and ISO C
3. C99
4. C11
5. C18
6. Embedded C

Syntax

C has a very corporate and formal grammar set by the C standard. Line endings are collectively not that meaningful in C. Although, line boundaries do have an importance during the preprocessing stage. Comments can display either between the delimiters /* and */, or (since C99) following // right until the line ending. Comments written by /* and */ do not nest, and these sequences of characters are not interpreted as comment delimiters if they appear inside string or character literals.

Character Set

The base C source character sets are the following characters:

1. Lowercase and Uppercase letters of ISO Basic Latin Alphabeth.
2. Decimal digits.
3. Special or Graphic characters
4. Whitespace characters.

Reserved Words

Here is a list of a few of the 32 reserved words that C has.

1. auto
2. break
3. case
4. char.
5. const
6. continue
7. default
8. do
10. double
11. else
12. if
13. enum
14. extern
15. float

Operators

C provides support for a large set of operators, which are symbols used within an expression to state the manipulations to be carried out while evaluating that expression. **C has operators for:**

1. Arithmetic: +, -, *, /, %
2. Assignment: =
3. Augmented assignment: +=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=
4. Bitwise logic: ~, &, |, ^
5. Bitwise shifts: <<, >>
6. Boolean logic: !, &&, ||
7. Conditional evaluation:? :
8. Equality testing: ==, !=
9. Calling functions: ( )
10. Increment and decrement: ++, --
11. Member selection: ., ->
12. Object size: sizeof
13. Order relations: <, <=, >, >=
14. Reference and dereference: &, *, [ ]
15. Sequencing: ,
16. Subexpression grouping: ( )
17. Type conversion: (typename)

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

C-SHARP PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is C# or C-Sharp?

C# is a simple, multi-purpose, modern object-oriented software programming language that is developed and managed by Microsoft within their .NET initiative led by **Anders Hejlsberg.**

C# was created and developed by Anders Hejlsberg and his software development team during the development period of the .Net Framework. C# is designed for the Common Language Infrastructure (CLI), which is made up of the executable code and runtime environment that allows for the use of other high-level object-oriented languages on different computer platforms and architectures.

The C# Programming Language, however, was developed with the needs of business and enterprises in mind. C# Language was developed for businesses to develop all kinds of software by using one single programming language.

Although C# constructs closely follow the pattern of traditional high-level languages, such as C and C++ and being an object-oriented software programming language. It has a strong resemblance to Java, it also has lots of powerful programming features that make it beloved and widely used by a number of programmers around the world.

Features Of C# Programming

There are many features of C# and some of them are:

1. Simplicity: All the Syntax of C# is just like C++. There is currently no preprocessor and it has a much larger library of reusable codes. C# code does not need you to include header files. All code in c# is written inline.

2. Consistent Behavior: C# brought about a unified type of system that solves and eliminates the problem of varying ranges of integer data types. All data types are treated as objects and software developers can further extend the data type system simply and easily.

3. Modern Programming Language: C# offers supports for a number of modem features, such as Error handling features, Automatic Garbage Collection, Robust Security features, Modern debugging features, etc

4. Pure Object-Oriented Programming Language: In C#, everything is treated as an object. There are no more globally defined functions, variables, and constants. It provides supports for all three object-oriented features such as Encapsulation, Inheritance, and Polymorphism.

5. Type Safety: Type safety brings about better and more robust programming. With type safety, all the objects and arrays are dynamically initialized by zero, an error message will be printed out, on use of a variable that is uninitialized, it also allows for the automatic checking of arrays.

6. Feature of Versioning: Releasing new versions of a software module and making it work with the existing applications is referred to as versioning. It is gotten by the keywords known as new and overrides.

7. Compatible with other Language: C#, makes sure that you follow all the .NET common language specifications (CLS) and therefore it allows for interoperation with other .NET languages.

8. Inter-operability: Language interoperability defined as the ability of a piece of code to interact with other codes that are written using a different software programming language. Language interoperability in c# can help to maximize code reuse and, therefore, improve the efficiency and performance of the development process.

Why Study C# Programming?

Below are some of the reasons of studying C# Language

1. Simple to Get Started With: C# was developed to be simple and very easy to use. Since C# is a high-level computer programming language, it somewhat reads closer to English. In addition, C# abstracts away (i.e. manages for you) most of the complex details of the machine (computer) so you can just focus on application programming logic instead of worrying about the little details that most people see to be both difficult and tedious.

2. Easy to Maintain: C# is a statically-typed software language, this means that your code will be checked for errors before it gets compiled into an application. Errors will be very easy to track down, and since statically-typed languages are also very strict with how you program something, the codebase, in general, will be further consistent and therefore easier to maintain as it grows in complexity and size.

3. Fast: As a statically typed language, C# is faster than most dynamically typed languages because things are much more clearly defined. Therefore, when the application is running, your machine's resources will not be wasted on checking for the definition of something in your code.

4. C# has the 4th Largest StackOverflow Community: StackOverflow is a software programming Q&A community site that you would no doubt become close with as a coding beginner. Since StackOverflow was built with C#, the C# community there is very strong. Some software developers think that C# questions get higher quality answers when compared to other languages.

5. Career Opportunities: 16% of the top 100 million websites in the world are powered by the ASP.NET framework, of which most of them might be using C#, so there are many job opportunities in enterprise-level backend development. With the emergence of indie game development, for example, C# developers are very much likely to have an additional opportunity, since Unity has pretty much become the de facto game engine for indie game development. Also, if you check on top job aggregator websites and boards, you would find a large number of C# jobs.

6. Future: As C# was designed and developed by Microsoft to build apps on the Microsoft platform, Microsoft will very likely make sure that they keep C# relevant and updated. A programming language's ability to stay consistent and survive also depends on whether the programming language is getting new users. In the are of search volume according to Google Adwords, C# has a large number of search volume and is the 5th place in terms of the programming language that people are most interested in learning.

7. Self-Employment Opportunity and Consultancy: You can be a freelancer or entrepreneuer as a C# developer or software developer.

Benefits Of C# Programming Language

There are Lots of benefits and advantages you get from learning and coding with C# and some of them are:

1. C# is an Object-Oriented Language.

2. C# has an Automatic Garbage Collection.

3. In C#, there is no problem of Memory Leak

4. C# brings eases the process of development, and its applications are Cross-Platform

5. Better Integration

6. More Legible and easy to understand Coding

7. C# has a huge support for whatever problem you encounter while coding.

C# Course Outline

C# - Introduction/Overview
C# - Environment
C# - Program Structure
C# - Basic Syntax
C# - Data Types
C# - Type Conversion
C# - Variables
C# - Constants
C# - Operators
C# - Decision Making
C# - Loops
C# - Encapsulation
C# - Methods
C# - Nullables
C# - Arrays
C# - Strings
C# - Structure
C# - Enums
C# - Classes
C# - Inheritance
C# - Polymorphism
C# - Operator Overloading
C# - Interfaces
C# - Namespaces
C# - Preprocessor Directives
C# - Regular Expressions
C# - Exception Handling
C# - File I/O
C# - Attributes
C# - Reflection
C# - Properties
C# - Indexers
C# - Delegates
C# - Events
C# - Collections
C# - Generics
C# - Anonymous Methods
C# - Unsafe Codes
C# - Multithreading
C# - Video Lectures
C# - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

D PROGRAMMING LANGUAGE COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is D Programming?

D Programming Language is a high-level object-oriented multi-paradigm system programming language. D programming was originally designed and developed by re-engineering the C++ programming language, but it is a distinct programming language that not only makes use of some features of C++ but also some features of other programming languages like Java, Python, C# and Ruby.

D Programming Language is a general-purpose programming language that possesses static typing, systems-level access, and C-like syntax. It is a higher-level language than C++ but it still retains the ability to write high-performance code and interface directly with the operating system's APIs and with the hardware.

D Programming Language is well designed for writing small to large scale programs with a team of developers. **D** is very easy to learn, and it provides many capabilities and features to aid the programmer in coding and is well suited to aggressive compiler management technology.

D Programming is not a scripting language, neither is it an interpreted language. It doesn't come with a Virtual Machine, a design philosophy, or an overriding belief. It's a more practical language for practical programmers who need to get the job done very quickly, reliably, and also making sure that the designed project is easily maintainable and easy to understand.

D Programming supports both automatic (garbage collected) and manual memory management. Its programs are essentially structured as components or modules that can be compiled separately from each other and then linked together with external libraries to build native libraries or executables

Features Of D Programming

D Programming has a lot of features and some of them are explained below:

1. Garbage Collection: Garbage collection (GC) is a constantly changing approach to the spontaneous memory management and heap allocation that processes and identifies dead memory blocks and reassigns storage for reuse. The primary purpose of garbage collection is to reduce leaks of memory in software.

2. Functions: A function is a block of code that is often defined by the role it will play within a larger code structure. Precisely, a function is made up of a unit of code that works on multiple inputs, many of which are stored as variables, and delivers accurate results concerning changes to variable values or actual operations that are based on the inputs.

3. Arrays: In programming, an array is a series of objects that all have the same size and type. Each object that is in an array is called an element of an array. For example, you could have an array of characters, or an array of integers or generally an array of anything that has a defined data type.

4. OOP: Object-Oriented programming, mostly referred to as OOP is a Computer Programming language design pattern in which the program is developed around data, or objects, instead of just writing functions and logic. An object can be defined as a data field that has individually unique attributes and behavior.

5. Performance: In computing, computer performance is the amount of useful work that is accomplished by a computer system. Outside of the specific contexts, computer performance is calculated in terms of efficiency, accuracy and the speed of executing computer program instructions.

6. Generic Programming: Generic Programming is defined as a Computer Programming design model whereby algorithms are designed and developed in terms of types to-be-specified-later that are then instantiated when needed for specific types presented as input parameters.

7. Memory Management: Memory is usually managed with the help of garbage collection, but very specific objects may be finalized immediately they go out of scope. Garbage collection can be monitored and controlled by programmers. They can achieve this by adding or excluding memory ranges from being noticed by the collector. They can also disable or enable the collector and force either a generational or full collection cycle.

8. Design by Contract
9. Unit Testing
10. True Modules
11. Automatic Memory Management
12. Cross-Platform
13. Operator Overloading
14. Documenting Comments
15. Exceptions
16. Inline Assembler
17. Mixins
18. Contracts

Benefits And Advantages Of D Programming

1. Multi-Paradigm System
2. Imperative
3. Object-oriented Programming
4. Meta-programming
5. Functional programming Style
6. Fast Compiling and Execution time
7. Better Memory Management
8. Easy Syntax phrasing
9. Fixed Data Knowledge and Pure
10. D is quick for prototyping.

Development Tools for D Programming

List of tools that can be used to ease the development of software with D programming language.

1. Eclipse plug-ins for D.
2. Visual studio code integration with Extensions as Dlang-vscode.
3. Vim.
4. Visual Studio.
5. TextMate
7. Code::Blocks
8. Xcode 3 using a plugin called D for xcode.
9. An Add-in for MonoDevelop named Mono-D.
10. KDevelop autocompletion plugin
11. Coedit, an open-source IDE dedicated to D

Benefits And Advantages Of D Programming

This simple program prints its command line arguments.

```
import std.stdio: writefln;

void main(string[] args)
{
    foreach (i, arg; args)
        writefln("args[%d] = '%s'", i, arg)
}
```

Why You Should Learn D Programming

1. Modeling Power: D Programming gives you the option to choose whatever method is convenient for modeling code to fit your idea.

2. Productivity: D Programming allows programmers to be more productive because you can do more in less time.

3. Readable and Maintainable: D Programming code is quite easy to understand by anyone familiar with C-type programming languages.

4. Convenient: D Programming eliminates the issue of frequently changing or having too many parts.

5. Job Opportunities and Career Advancement

D Programming Course Outline

D Programming - Introduction/Overview
D Programming - Environment
D Programming - Basic Syntax
D Programming - Variables
D Programming - Data Types
D Programming - Enums
D Programming - Literals
D Programming - Operators
D Programming - Loops
D Programming - Decisions
D Programming - Functions
D Programming - Characters
D Programming - Strings
D Programming - Arrays
D Programming - Associative Arrays
D Programming - Pointers
D Programming - Tuples
D Programming - Structs
D Programming - Unions
D Programming - Ranges
D Programming - Aliases
D Programming - Mixins
D Programming - Modules
D Programming - Templates
D Programming - Immutables
D Programming - File I/O
D Programming - Concurrency
D Programming - Exception Handling
D Programming - Contract
D Programming - Conditional Compilation
D Programming - Object Oriented
D Programming - Classes & Objects
D Programming - Inheritance
D Programming - Overloading
D Programming - Encapsulation
D Programming - Interfaces
D Programming - Abstract Classes
D Programming - Video Lectures
D Programming - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

DATA STRUCTURE AND ALGORITHMS COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Data Structure & Algorithms

A Data Structure is a specialized method or format for organizing and storing data. General data structure types include:

1. The array,
2. The file,
3. The record,
4. The table,
5. The tree, and so on.

Algorithms are set of rules and processes to be followed by the computer to carry out various calculation tasks and problem-solving operations.

All Data Structures are designed to organize data in order to suit a specific purpose so that it can be accessed and worked with in appropriate ways. In computer programming, a data structure may be selected or designed to store data for the purpose of working on it with various algorithms.

Data Structure is a systematic way to organize data for effective usage. The following are the fundamentals of a data structure.

1. Interface: This represents the set of operations data structure supports. However, an interface provides the list of supported operations, parameters.

2. Implementation: This also provides the internal representation of a data structure. It provides the definition of the algorithms used in the operations of the data structure.

3. Need for Data Structure: Data can be organized or reorganized in a data structure in a way that not all items may not be required to be searched and the required data can be searched instantly.

4. Data Queue: This is a process whereby one end is used to input data (enqueue) and the other is used to remove data (dequeue). It can as well be described as (FIFO) First-In-First-Out methodology. Which means the first data item stored will be the first to be accessed.

5. Merge Sort: This is a technique that is used to divide array into equal halves and then combines them in a sorted manner.

6. Selection Sort: This is a method of sorting. It is an in-place comparison-based algorithm which the list is however divided into two parts.

(a) Sorted part and
(b) Unsorted part.

The **Sorted part** is on the left side while the **Unsorted part** is on the right side.

7. Binary Search: This is a fastest way of conducting a search. What applies is a principle of divide and conquer and in order for the algorithm to work properly, the data collection must be in a sorted form.

Advantages of Data Structure & Algorithms

1. Data structures allow information storage on hard disks.

2. Data structures allow the data use and processing on a software system.

3. Provides means for management of large data-set such as databases or internet indexing services.

4. It allows easier processing of data.

5. It allows safe storage of information on a computer. The information is then available for later use and can be used by multiple programs.

In the Full course, you will learn everything you need to know about **Data Structure and Algorithms** with **Certificate** to showcase your knowledge.

Data Structure and Algorithms Course Outline

Data Structures & Algorithms - Introduction/Overview
Data Structures & Algorithms - Environment Setup
Data Structures & Algorithms - Algorithms Basics
Data Structures & Algorithms - Asymptotic Analysis
Data Structures & Algorithms - Greedy Algorithms
Data Structures & Algorithms - Divide and Conquer
Data Structures & Algorithms - Dynamic Programming
Data Structures & Algorithms - Data Structure Basics
Data Structures & Algorithms - Array Data Structure
Data Structures & Algorithms - Linked List Basics
Data Structures & Algorithms - Doubly Linked List
Data Structures & Algorithms - Circular Linked List
Data Structures & Algorithms - Stack
Data Structures & Algorithms - Expression Parsing
Data Structures & Algorithms - Queue
Data Structures & Algorithms - Linear Search
Data Structures & Algorithms - Binary Search
Data Structures & Algorithms - Interpolation Search
Data Structures & Algorithms - Hash Table
Data Structures & Algorithms - Sorting Algorithms
Data Structures & Algorithms - Bubble Sort
Data Structures & Algorithms - Insertion Sort
Data Structures & Algorithms - Selection Sort
Data Structures & Algorithms - Merge Sort
Data Structures & Algorithms - Quick Sort
Data Structures & Algorithms - Graph Data Structure
Data Structures & Algorithms - Depth First Traversal
Data Structures & Algorithms - Breadth First Traversal
Data Structures & Algorithms - Tree Data Structure
Data Structures & Algorithms - Tree Traversal
Data Structures & Algorithms - Binary Search Tree
Data Structures & Algorithms - AVL Tree
Data Structures & Algorithms - Spanning Tree
Data Structures & Algorithms - Heap
Data Structures & Algorithms - Recursion Basics
Data Structures & Algorithms - Tower of Hanoi
Data Structures & Algorithms - Fibonacci Series
Data Structures & Algorithms - Video Lectures
Data Structures & Algorithms - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

FLUTTER COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Flutter?

Flutter is a free and open-source mobile SDK that enables developers to create native-looking Android and iOS applications from a single code base. Flutter has been around since 2015, when Google introduced it, and remained in beta until December 2018, when it was officially launched. Since then, the buzz surrounding Flutter has intensified.

Flutter has surpassed the top 11 software repositories on GitHub in terms of stars. Additionally, we've seen thousands of Flutter apps published to app stores. Among the most notable examples is the Alibaba team's Xianyu app, which is used by over 50 million people.

How Does Flutter Work?

Flutter's central concept is the use of widgets. Developers construct the entire user interface by combining various widgets. Each of these widgets defines a structural element (for example, a button or a menu), a stylistic element (for example, a font or color scheme), and a layout aspect (for example, padding).

Notably, Flutter does not use OEM widgets but instead provides developers with pre-built widgets that appear to be native to Android or iOS apps. Developers, of course, are free to create their own widgets.

Likewise, Flutter provides reactive-style views for developers. Flutter uses Dart as the JavaScript bridge language to avoid performance issues associated with using a compiled programming language. It performs asynchronous compilation of Dart into native code for multiple platforms.

Thus, Flutter can communicate with the platform directly without the need for a JavaScript bridge that requires a context switch between the JavaScript and native realms. As you might expect, compiling to native code reduces the startup time of the application.

Flutter is the only mobile SDK available today that supports reactive views without requiring a JavaScript bridge. That is why a large number of mobile developers have used it in their projects.

Flutter's use of the Dart language is one of its most intriguing features. Flutter, like other systems that make use of reactive views, updates the view tree with each new frame. To accomplish this, it generates a large number of objects that have a maximum lifespan of one frame. Dart employs generational garbage collection, a technique that has been shown to be extremely efficient in this type of system.

Also, Dart includes a "tree shaking" compiler that generates only the code necessary for your application. Even if you only require one or two widgets, you can freely access the library's vast collection.

Finally, Dart includes a repository of software packages that enable developers to extend the capabilities of their applications. For instance, it provides a few packages that enable developers to connect to Firebase and create serverless applications. Another package simplifies the process of accessing a Redux data store or platform services and hardware such as the camera.

Benefits Of Flutter

There are many benefits associated with using Flutter, let's take a look at some of them:

- Compatibility
- Rapid Development
- Cost effective
- Time Saving
- Excellent Performance
- Open-source

Compatibility

Flutter comes with its own widgets, which means that compatibility issues are minimized. Developers encounter fewer issues when testing the app across multiple OS versions and will save time testing the app on older OS versions. Also, you can rest assured that your application will continue to function on future OS versions.

Note: When a new version of Android or iOS is released, Flutter widgets must be updated (as the tool does not make use of native platform widgets). You may be wondering when Flutter's widgets will be updated.

Given that Google is a significant internal user of Flutter, the Flutter team is highly motivated to keep their widget sets as current and as close to the platform widgets as possible. Additionally, Flutter widgets are completely customizable and updateable by anyone. Even older OS versions can run your app with new widgets!

Rapid development is possible due to the use of hot reloading

Due to the fact that Flutter supports hot reloading, it is gaining significant traction among mobile developers. On emulators, simulators, and hardware, hot reload enables instantaneous viewing of the changes made to the code. Within a millisecond, the modified code is reloaded. The app continues to run throughout, and developers are spared the time-consuming task of restarting it.

This facilitates the development of user interfaces, the addition of new features, and the correction of bugs. When an app encounters an error, it is typically possible to resolve it and then continue using the app as if nothing happened. Even if you are forced to perform a complete app reload, you can rest assured that it will be completed quickly, thereby speeding up the development process.

It is cost-effective and time-saving

Flutter is a tool for developing cross-platform applications. This means that developers can use the same code base to create iOS and Android applications. Cross-platform development is the most efficient way to conserve time and resources during the development process.

Excellent performance

Flutter performs exceptionally well in two ways. To begin, it makes use of Dart, a programming language that compiles to native code. Second, because Flutter ships with its own widgets, there is no need to use OEM ones. As a result, the app and the platform communicate less effectively. These two Flutter features contribute to faster app startup times and fewer overall performance issues.

Open-source

Flutter is an open-source technology with a thriving developer community that provides support, contributes to the tool's extensive documentation, and creates useful resources. Dart and Flutter are both open-source applications.
Flutter is a cutting-edge mobile technology that is currently available on the market. Due to the benefits it provides development teams, it is a likely candidate to become the preferred mobile technology in the near future.

Flutter Course Outline

Flutter - Introduction
Flutter - Installation
Flutter - Creating Simple Application in Android Studio
Flutter - Architecture Application
Flutter - Introduction to Dart Programming + Flutter
Flutter - Introduction to Widgets
Flutter - Introduction to Layouts
Flutter - Introduction to Gestures
Flutter - State Management
Flutter - Animation
Flutter - Writing Android Specific Code
Flutter - Writing IOS Specific Code
Flutter - Introduction to Package
Flutter - Accessing REST API
Flutter - Database Concepts
Flutter - Internationalization
Flutter - Testing
Flutter - Deployment
Flutter - Development Tools
Flutter - Writting Advanced Applications
Flutter - Video Lectures
Flutter - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

FORTRAN COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Fortran?

Fortran is an imperative programming language compiled for general purposes that are particularly suitable for numerical computing and scientific computing.

Fortran was first launched in 1954, Fortran is and is still commonly used as the oldest programming language. Scientific fields, in particular, numerical weather prediction, computational fluid dynamics, and computational physics find their applications.

Fortran is quite common in high-performance computing and is used to mark the program bench and rank the fastest supercomputers in the world.

Fortran is a compiled language, or more specifically it is compiled ahead-of-time. In other words, you must perform a special step called the compilation of your written code before you are able to run it on a computer. This is where Fortran differs from interpreted languages such as Python and R which run through an interpreter which executes the instructions directly but at the cost of computing speed.

Why Learn Fortran?

Fortran is the dominant programming language used in engineering applications. It is therefore important for engineering graduates to be able to read and modify Fortran code. From time to time, so-called experts predict that Fortran will rapidly fade in popularity and soon become extinct. These predictions have always failed. Fortran is the most enduring computer programming language in history. One of the main reasons Fortran has survived and will survive is software inertia.

Over the years, a vast collection of code has been written with Fortran in many scientific and engineering machines, allowing the language to persist.

Some of the reasons you should learn Fortran:

1. Built-in support for arguments in subroutines
2. Rich set of intrinsic functions
3. Built-in support for complex numbers
4. Support for array notation allowing operations on array sections
5. Strong aliasing rules for memory pointers, resulting in more efficient code after compilation
6. Some legacy code are written in Fortran and still in operation.

Fortran Basics

FORTRAN was designed for scientists, engineers, and programmers. There are a set of rules governing Fortran code and whether a code is valid and can be compiled. Fortran uses a popular code structure term called Syntax- a combination of characters and reserved key-words to create a Fortran program and issue an instruction to the computer.
Below are Basics of Fortran Programming:

1. Program Structure

This is the structure of the Fortran program and it has a reserved keyword for doing just that. Fortran uses the **PROGRAM** keyword, followed, optionally, by a name for the program to begin a program and end it with the **END PROGRAM** keyword.

Example:

```
PROGRAM myprogram
! Do some stuff, other code, statements, functions
END PROGRAM
```

Within the **PROGRAM** statements, you can define functions, declare variables to be used in these functions, just like in other programming languages.

2. Variables Definition:
These are values or data used in your program. A variable can refer to a single value like 1 or point to a larger data structure such as a list. They can also be used to store data like texts, words, decimal and many more.
Each variable usually refers to a type of data. A data type describes what kind of data the variable is expected to hold. It can either numbers which are integers or words which is a string. We could say **x=3,** this means **x** is holding an integer type of data, however, we didn't specify what type of data it's holding. Most programming languages require you to specify what type of data a particular variable is expected to hold.

In Fortran, variables are declared as below:

```
INTEGER :: n = 3
INTEGER :: m = 6
```

The above code in Fortran means we are creating two **INTEGER** variables. The keyword **INTEGER** is used to specify what kind of data we are creating and that's a variable with numbers. There are others like **REAL, LOGICAL,** etc.

3. Read and Write Message:

This is the easiest way to see the results of a calculation or print a message on the console and also collects inputs from users to be used in a calculation for instance. This basically provides a way to print a message on the console and a way for users to type in values using the keyboard right from the program.

Read: The READ function tells the FORTRAN program to record the values you enter using the keyboard and store them in variables that you define in your program. We can see a working example :

```
READ(*,*) a, b, c
```

In Fortran functions, any inputs needed for the function are placed inside a list within the two round-brackets (parentheses), as in reading (input1, input2, etc…). When the program is running and gets to the READ statement, it will prompt you to enter a variable and press enter.

Write: WRITE function is similar to the **READ** function. It just prints out the given values to the console screen. For instance:

```
WRITE(*,*) a, b, c
```

This would print the values assigned to the variables specified in the code.

4. Mathematical Operations:

Fortran is primarily designed for numerical calculations, and it has many built-in functions for mathematical operations. In Fortran, we use the basic functions: +, -, *, / for addition, subtraction, multiplication, and division.

5. Decision Making:

This is the logic flow of a program. This is for instance where most calculations are being made and decides on the final results. It is a bit like the logical decisions we make in our everyday life. For instance, we could say **IF** it's raining **THEN** don't go outside, **ELSE** there is an umbrella you are free to go to. Below is an example:

```
LOGICAL :: Raining
LOGICAL :: Umbrella
IF (Raining .AND. Umbrella) THEN
     WRITE(*,*) "Its raining and I have an umbrella, I can go out"
ELSE
     WRITE(*,*) "Its raining and I have no umberella, I can't g out"
END IF
```

6. Components of Decision Making:

The IF and THEN statements: These two keywords are used to make logical decisions within a program. The IF [CONDITION(S)…] THEN construct is used to do something based on a condition being met.
The ELSE statement: The ELSE statement is usually optional, although it's useful if you have something you want to do if the condition is false. Most times, the ELSE statement is used here to print an error message if the conditions are not met.
End IF statement: This indicates the end of a decision-making block. You must remember to always end it with the END IF statement.

7. Compilation:

Most programming languages usually need to be compiled or go through a process before it can be run or executed. Fortran programs need to be compiled to what the computer understands using the gfortran compiler. This compiles Fortran code to what the computer understands. Save your code in a text file with .f90 extension and run it against the compiler to generate the machine code.

You can run the command below in your terminal directory:

```
gfortran example.f90 -o example
```

This tells gfortran to take the file example.f90 and generate a program called example for us to run. The **-o** flag lets us indicate an output name for our generated program. This will generate a file example in your directory which can be run from the command line.

Advantages of Fortran

Fortran has many advantages, some of which are:

1. In science circles, Fortran is very powerful. It has a wealthy tool ecosystem for this domain.

2. Fortran produces the world's fastest native code. That's why scientific computing is perfect.

3. Fortran is extremely optimized for vectorization, making it great for modeling supercomputers.

4. Modern Fortran is amazingly simple to read and comprehend. Compare to C++.

5. Fortran does not use the terrible syntax of the off-side rule you find in Python.

Main Features of Fortran

1. FORTRAN II endorsed the language of procedural programming that enables algorithmic writing of the code.

2. The COMMON declaration is used to access the global variables to be used in the programs for subroutines.

3. FORTRAN II enables the subroutines to perform the tasks and the return values to be written.

4. Any inherent or derived type may be a POINTER.

5. Lower case characters in the source text are now permitted. A semicolon can be used in a single source line to distinguish various statements.

Fortran Course Outline

Fortran - Introduction/Overview
Fortran - Environment Setup
Fortran - Basic Syntax
Fortran - Data Types
Fortran - Variables
Fortran - Constants
Fortran - Operators
Fortran - Decisions
Fortran - Loops
Fortran - Numbers
Fortran - Characters
Fortran - Strings
Fortran - Arrays
Fortran - Dynamic Arrays
Fortran - Derived Data Types
Fortran - Pointers
Fortran - Basic Input Output
Fortran - File Input Output
Fortran - Procedures
Fortran - Modules
Fortran - Intrinsic Functions
Fortran - Numeric Precision
Fortran - Program Libraries
Fortran - Programming Style
Fortran - Debugging Program
Fortran - Video Lectures
Fortran - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

F# PROGRAMMING LANGAUGE COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is F# or F-Sharp?

F# Programming is a multi-purpose, multi-paradigm, strongly typed, software programming language that encompasses both functional, imperative, and object-oriented programming designs. F# is mostly used as a cross-platform Common Language Infrastructure (CLI) language, but it can also be used to generate JavaScript and GPU (graphics processing unit) codes.

F# Programming is designed and developed by the F# Software Foundation, Microsoft and other open contributors. An open-source, cross-platform compiler for F# is made available from the F# Software Foundation.

What is F# or F-Sharp?

F# Programming is a fully supported language in the following development tools:
1. Microsoft Visual Studio
2. Xamarin Studio
3. Mono,
4. MonoDevelop,
5. SharpDevelop,
6. MBrace, and
7. WebSharper.

Plug-ins supporting F# are found in many widely used code editors, most notably the Ionide extension for Atom and Visual Studio Code, and its integrations for other editors such as Vim, Sublime Text, Emacs, and Rider

F# is a member of the ML language family and it came up as a .NET Framework implementation of a core of the programming language **OCaml.** It has also been greatly impacted by languages like C#, Python, Scala, Haskell, and Erlang.

Functional Programming handles computer programs like mathematical functions. In functional programming, the main focus would be on the constants and functions, instead of just on variables and data states. Because functions and constants are things that don't change throughout the code. Also In Functional programming, you will write modular programs, for example, the programs would be made up of reusable functions that can take other functions as its input.

Why Study F#

There are many reasons to study F# and some of them are:

1. Ease of learning, much simpler and easy to understand code syntax.
2. F# has a very powerful development environment for Windows clients and other server-side applications.
3. F# is a functional software programming language, this feature ensure that applications you write is modular and extensive.
4. Job opportunity and career advancement.

Features of F# Programming Language

There are lots of features of F# and some of them are:

1. Conciseness: F# presents software developers with a clean and nice code syntax to write no curly brackets, no semicolons and so on as seen in most languages. Also, you don't have to specify the data type in your code just because of type inference. Moreover, you can finish your code in a lesser amount of line as compared to other programming languages.

2. Convenience: Common programming tasks are very much simpler to implement in F#. You can easily structure and process your complex code problems. Since functions are its first-class object so it is very easy to create a powerful and reusable code module by defining functions that make use of other functions as its parameter.

3. Correctness: F# comes with a powerful data type system that helps to deal with commonly found in programming such as null reference exception, etc. F# is a strongly typed language and this helps developers to write error-free code. It is easily caught at compile time as a type error so the developer would quickly fix it as the code would not compile.

4. Concurrency: F# comes with a number of built-in functions and libraries to handle the programming system when multiprocessing has occurred. F# also offers support for asynchronous programming, message queuing system, and various modules to support event handling. Data stored in F# is immutable by default so sharing of data is very safe. It avoids lock during the communication of code.

5. Completeness: F# is a functional programming language but it also offers support for other programming approaches such as object-oriented approach, imperative, etc. these make it easier to communicate with other domains. Essentially, we can say that F# is designed to be a hybrid programming language by which you can do almost everything that you can do with other programming languages such as C#, Java, etc.

Benefits of F# Programming Language

There are lots of benefits and advantages of learning F# and some of them are:

1. F# is a simple language that is very easy to learn and easily accessible to most software developers.

2. F# has a strong development environment for Windows clients and various server applications.

3. Programmers who know F# are generally considered to be well above average.

4. F# is an interactive language, these means that its code runs immediately in the IDE. this makes F# ideal to be used for testing and scripting.

5. F# implements static type checking and this helps to catch lots of errors at compile time.

6. F# has an automatic generalization which means that functions are automatically made generic by the compiler whenever it is possible.

7. F# is a functional programming language, this makes code to be easily reused.

F# Programming Course Outline

F# - Introduction/Overview
F# - Environment Setup
F# - Program Structure
F# - Basic Syntax
F# - Data Types
F# - Variables
F# - Operators
F# - Decision Making
F# - Loops
F# - Functions
F# - Strings
F# - Options
F# - Tuples
F# - Records
F# - Lists
F# - Sequences
F# - Sets
F# - Maps
F# - Discriminated Unions
F# - Mutable Data
F# - Arrays
F# - Mutable Lists
F# - Mutable Dictionary
F# - Basic I/O
F# - Generics
F# - Delegates
F# - Enumerations
F# - Pattern Matching
F# - Exception Handling
F# - Classes
F# - Structures
F# - Operator Overloading
F# - Inheritance
F# - Interfaces
F# - Events
F# - Modules
F# - Namespaces
F# - Video Lectures
F# - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

GO PROGRAMMING LANGUAGE COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Go Programming Language?

Go Programming Language known as GoLang is a compiled programming language of general use developed by Google Inc. It was first created by Robert Griesemer, Rob Pike, and Ken Thompson at Google in 2007. It offers integrated support for garbage collection, memory safety, and structural typing.

GoLang is one of the languages that has come out of obscurity and now gaining popularity for various development tasks. Go is powering an extensive number of popular projects and supporting the development of top-notch innovations across several industries.

Features of Go Programming Language

Language Design: Golang is designed in a way that the language is simple and easy to understand and write. Though the learning steep curve is higher than a language like Python, however, it's worth it with its awesome features. They did some interesting design decisions while working on the Object-Oriented support in the language. The language is opinionated and recommends an idiomatic way of achieving simple and complex things. It prefers Composition over Inheritance and "Do More with Less" is the mantra.

Package Management: Go merges modern-day developer workflow of working with Open Source projects and includes that in the way it manages external packages. Support is provided directly in the tooling to get external packages and publish your own packages in a set of easy commands.

Powerful standard library: Go has a powerful standard library, which is distributed as packages.

Static Typing: Golang is a statically typed language which means its strict on its types. This not just work on compiling the code successfully but also ensures error-free type conversions and compatibility. With this feature, you can avoid all problems faced with dynamically typed languages.

Testing Support: Golang has extensive support for unit testing features natively. It's a simple mechanism to write your unit test parallel with your code because of this you can understand your code coverage by your own tests. This also helps in generating your code documentation as an example easily.

Platform Independent: Go language is just like Java language as it supports platform independence. Due to its modular design and modularity i.e., the code is compiled and is converted into binary form which is as small as possible and hence, it requires no dependency. Its code can be compiled in any platform or any server and application you work on.

Applications of Go Programming Language

Go Programming Language: can be applied in various aspects of technology and can be used for diverse ways in programming.

Web Development: Go can be used extensively in full fledge web development. The Go programming language has native support for Network/HTTP protocol which gives us the ability to create web applications using the language.

Command Line Interface: Go can be used to build programs that can be run right from the command line, shells or terminal. With the I/O features, you can easily build simple command-line applications. Go has various libraries that made this easier.

Networking Development: Go has a lot of features that are able to power the development of high-level networking tasks. With goroutines, channels, go innate speed, you can easily build and deploy complex networking software for powering modern tech demand.

Real-time Development: With Go programming language, you can build real-time applications. You can create programs that users sense as immediate or current, providing solutions for e-commerce, chatting, online gaming and more.

Cloud Infrastructure: Golang gives on-demand access to shared computer processing, allowing you to process your data on a privately-owned cloud or a third-party server. Cloud Platforms like Google Cloud and Docker platform were developed using Golang.

Microservices: A service-oriented architecture that structures an application, decomposes it to smaller services and makes it easier to understand, develop and test. Go-kit provides specialized support, such as infrastructure integration and system observability perfectly suited to writing microservices.

Advantages of Go Programming Language

It's Easy to Understand: In a matter of a few weeks, you can master it, sometimes even hours or days if you use the standard library. It's a piece of cake to write an easy command-line app and/with a straightforward HTTP server. It is concise, simple and easy to read.

Naturally Open Source: If there are any problems, Gophers will detect and correct the issues, as the Golang developers are called, so they are easily solved before they become a problem.

Concurrency: Go concurrency features allows multiple processes running simultaneously and effectively.

Garbage Language Gathered: The language facilitates the development of operations such as memory management for designers. It handles all the jobs of running on its own.

Type Conversion: This is actually a very good advantage because it makes your program type error-free. It validates for the interface and type embedding.

Go Language is Swift: It offers easy structure and syntax, so developers find it easy to learn. This makes it easy for you to rapidly and even better keep a product on the market.

Enhanced Testing Tool: Developers enjoy the fact that built-in testing tools and the fact that it has the easiest API come with the language. The effectiveness of these tools makes executable codes simple to test and profile.

Go Programming Language Benefits

Some of the most significant benefits of Go Programming are mentioned below:

1. Environmental support that adopts comparable patterns to dynamic languages.
2. Built-in concurrency support
3. The time to compile is fast
4. Go programs are easy, concise and secure
5. Production without external dependencies of statically linked indigenous binaries

Go Programming Language Course Outline

Go Programming Language - Introduction/Overview
Go Programming Language - Environment Setup
Go Programming Language - Program Structure
Go Programming Language - Basic Syntax
Go Programming Language - Data Types
Go Programming Language - Variables
Go Programming Language - Constants
Go Programming Language - Operators
Go Programming Language - Decision Making
Go Programming Language - Loops
Go Programming Language - Functions
Go Programming Language - Scope Rules
Go Programming Language - Strings
Go Programming Language - Arrays
Go Programming Language - Pointers
Go Programming Language - Structures
Go Programming Language - Slice
Go Programming Language - Range
Go Programming Language - Maps
Go Programming Language - Recursion
Go Programming Language - Type Casting
Go Programming Language - Interfaces
Go Programming Language - Error Handling
Go Programming Language - Video Lectures
Go Programming Language - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

IONIC COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Ionic?

Ionic is an open-source software development framework that is used for developing mobile applications. Ionic provides various tools and services for building and developing Mobile UI that will have the look and feel of a native application. The Ionic framework uses a native wrapper to be able to execute on mobile devices.

Ionic is a front-end HTML5 framework that is built on top of Cordova and AngularJS and it is targeted at building hybrid applications. To make it easier to understand, think of Ionic as the front-end User Interface framework that handles all the looks, feel and UI interactions your application needs to be both compelling and unique. it is kind of like the "Bootstrap for Native", but with the support for a vast range of common native mobile application components, nice and slick animations together with a beautiful design.

The Ionic Framework is aimed at enhancing the frontend user experience, such as (UX) User Experience and the (UI) User Interaction of an application like the controls, interactions, gestures, buttons and animations. Ionic is very easy to learn and it integrates nicely with other development libraries or frameworks, such as Angular, and can be used separately without the inclusion of any frontend framework.

Features Of Ionic

The following are the most important features of Ionic:

1. AngularJS: Ionic as a framework uses the AngularJS MVC architecture for building and managing rich single-page applications (SPA) that are optimized for mobile devices.

2. CSS components: With the native look and feel that comes with Ionic, these components offer's you almost all the elements that a mobile application depends on. The elements' default styling can be easily modified to fit your own designs.

3. JavaScript components: These mobile components are extending the CSS components with JavaScript functionalities to satisfy and cover all the mobile elements that cannot be accomplished only with HTML and CSS.

4. Cordova Plugins: The Apache Cordova plugins offer API services that are needed for using native device functions together with JavaScript code.

5. Ionic CLI: Ionic CLI is NodeJS utility that is being powered with various commands for starting, building, operating and emulating Ionic mobile applications.

6. Ionic View: Ionic View is a very useful platform for uploading, sharing and testing your mobile application on native mobile devices.

7. Simplicity: Ionic Framework is built with form start with simplicity in mind so that building Ionic applications is enjoyable, very easy to learn, and accessible to just about anyone who has web development skills.

8. License: The Ionic Framework is a free and open-source project, and it is released under the permissible MIT license.

Benefits Of Ionic

The following are some of the most commonly known advantages of Ionic:

1. Ionic is used for the development of Hybrid Applications. This simply means that you can bundle your applications for both IOS, Android, Windows Phone and the Firefox OS, which can save you a lot of development time.

2. Starting an application with Ionic is very easy since it provides you with a useful pre-generated app setup with simple and basic layouts.

3. The applications are developed in a very clean and easily scalable way, so they are very maintainable and easy to update.

4. The Ionic Developers Team has a very good relationship with the Google Developers Team and they are constantly working together to improve the framework.

Ionic Course Outline

Ionic - Introduction/Overview
Ionic - Environment Setup
Ionic - CSS Components
Ionic - Colors
Ionic - Content
Ionic - Header
Ionic - Footer
Ionic - Buttons
Ionic - Lists
Ionic - Cards
Ionic - Forms
Ionic - Toggle
Ionic - Checkbox
Ionic - Radio Button
Ionic - Range
Ionic - Select
Ionic - Tabs
Ionic - Grid
Ionic - Icons
Ionic - Padding
Ionic - Javascript Components
Ionic - Cordova Integration
Ionic - AdMob
Ionic - Camera
Ionic - Facebook
Ionic - In App Browser
Ionic - Native Audio
Ionic - Geolocation
Ionic - Media
Ionic - Splash Screen
Ionic - Video Lectures
Ionic - Exams And Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

PROGRAMMABLE LOGIC CONTROLLERS (PLC) COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Programmable Logic Controllers (PLC)?

A Programmable Logic Controller sometimes abbreviated to as a **PLC or a Programmable Logic Controller** is an industrial digital computer that has been tested and tried and fully adapted for the control of manufacturing processes, such as robotic devices or assembly lines, or any activity that requires high-reliability control and ease of programming and process fault study and diagnostics.

Programmable Logic Controllers were first developed in the automobile manufacturing industry to support more resilient, ruggedized and efficient programmable controllers to be used to replace hard-wired timers, relays, and sequencers. Since then, they have been broadly adopted as high-reliability automation controllers that are suitable for harsh environments.

A Programmable Logic Controller is an example of a real-time system since the output that is produced must be in response to the input conditions within a limited amount of time, with which an otherwise operation will result.

Programmable Logic Containers can range from small and modular devices that have tens of
inputs and outputs (I/O), with a processor in a housing integral, to a large rack-mounted modular device that has a count of thousands of I/O, and which are mostly networked to other SCADA and PLC systems.

Features of Programmable Logic Controllers?

There are many features of Programmable Logic Controllers and some of them are:
1. Application programming: Soft Programmable Logic Controllers support loadable functions that are developed by users who are developers in C, C++ or Java. They also support online run mode program modifications, different documentation modes of programs, and I/O forcing. Soft Programmable Logic Controllers are developed with at least 6 different types of programming languages to make sure that they provide a wide range of online development, troubleshooting, and documentation features as well as to run converted or imported programs.

2. Capacity: Soft Programmable Logic Controllers systems have the ability to regulate over 16K I/O which is made up of digital, analog and specialty I/O. You can add up to 16 different I/O systems on a single soft Programmable Logic Controller regulator; it can have up to 36 COM ports that are linked to serial devices for ASCII communications and 32 user-configurable data points and networks as main communication mediums that allow data access from several computer PLCs or software applications. Also, soft Programmable Logic Controllers can hold large application programs and 100 million data table elements.

3. Hardware and operations: A soft Programmable Logic Controller works as an embedded 32-bit multi-tasking OS kernel that carries out its duties in real-time on different computer CPUs, including the x86 compatible CPU system platforms. It is independent of hardware busses like ISA, PC/104, VME, and PCI. It works under a more minimal hardware requirement, which makes sure of high reliability at very low costs.

Benefits of Programmable Logic Controllers

There are lots of benefits of Programmable Logic Controllers and some of them are;

1. It allows for the ease of development in the manufacturing firm.

2. It has High reliability and minimal maintenance feature.

3. Programmable Logic Controllers have a smaller physical size.

4. PLC has the ability to communicate directly with computer systems in the manufacturing firm.

5. PLCs have a moderate to low initial investment cost.

6. PLCs are made available in modular designs.

7. Job opportunities and career advancement.

8. Enrich your CV and increase your earning potential.

Programmable Logic Controllers Course Outline

PLC - Introduction
PLC - Input-output Devices
PLC - Digital Systems
PLC - I/O Processing
PLC - Ladder and Functional Block Programming
PLC - IL, SFC and ST Programming Methods
PLC - Internal Relays
PLC - Jump and Call
PLC - Timers
PLC - Counters
PLC - Shift Registers
PLC - Data Handling
PLC - Designing Programs
PLC - Programs
PLC - Video Lectures
PLC - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

IOS COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is iOS?

iOS is a Mobile Operating System (OS) that is developed and distributed by Apple Inc.

iOS was originally released in 2007 for the iPhone, Apple TV, and iPod Touch. iOS is gotten from OS X, from which it shares the Darwin foundation. iOS is Apple's mobile version of the OS X software operating system that is used in Apple computers.

Mobile devices that run on iOS get upgraded quite frequently and from experience, we find out that at least one version of the iPhone and iPad is released every year. currently, we have the iPhone 11 that was recently launched which has its predecessors Similarly, the iPad has evolved from iPad (1st Generation) to a more recent version which has evolved greatly together with other versions of iPad Mini.

The iOS SDK has greatly evolved from 1.0, the latest iOS SDK is the only officially supported version in Xcode 4.5 and higher. We have a rich Apple documentation with which we can find the various methods and libraries that can be used based on our deployment target in iOS development. In the latest version of Xcode, software developers will be able to choose between various deployment targets of all the supported iOS versions.

The number of users that are using the iPhone/iPad has increased a great deal since 2007. This alone opens up a great opportunity for software developers to make money by creating applications for iPhone and iPad and deploying it to Apple's App Store.

Apple Inc. also designed an online store known as: App Store where users can buy apps that have been developed for their iOS devices. A developer can deploy both free and paid apps to the App Store. To develop an application and distribute it to the store, the software developer will be required to register with the iOS developer program which will cost them approximately $99 a year and a Mac with Mountain Lion installed or higher for its development with the latest version of Xcode.

Features of IOS

There are many features of iOS that makes it very popular and really important to study, and some of them are:

1. Dark Mode.
2. Photos.
3. Camera.
4. Privacy and Security.
5. Sign in with Apple.
6. HomeKit.
7. Maps.
8. AirPods.
9. Memoji and Messages.
10. Reminders.
11. Performance.
12. Augmented Reality.
13. Accessibility.

Benefits of IOS

There are lots of benefits of iOS and below are some of them:

1. iOS Provide Very High Security: While developing an app for a business, a company has to choose a platform with various exciting features and the platform should also provide them with the utmost level of security to their users. iPhone users always stay safe from external threats. iOS offers a strong shield for viruses and malware, making it the perfect choice for app development of any business.

2. Secure Transactions: Though security measures are always implemented to enhance the security aspect of an online payment gateway, iOS makes it even more secure for its users by encrypting the online deals and removing various threats like phishing, SQL-injection, etc. The market image of iOS and its supportive devices such as the iPad and iPhone are making it increasingly important to improve business sales and productivity.

3. Nice User Interface: iOS applications are known to have a very nice, intuitive, and sleek User Interface(UI) so as an iOS developer, you are offered very unique and sleek app components to ensure that your applications all look awesome.

Career Opportunities in IOS

An increasing number of companies are bringing up the need for iOS professionals. If you are in the look for career opportunities that are available in this field, they are large, and some of them are listed below.

1. You get a lot of job opportunities with iOS Development.

2. You can become an iOS Development Specialist in an organization.

3. After taking this certification course of the IOS Development module, you can become a Certified Mobile Application Developer or Engineer.

4. You can become a Team Leader in any organization and be responsible for managing mobile user experience design.

5. You can become an iOS Development Specialist and Mobile Development Engineering Trainer.

6. You can become a Senior iOS Engineer commanding high pay.

IOS Course Outline

iOS - Getting Started
iOS - Environment Setup
iOS - Objective-C Basics
iOS - First iPhone Application
iOS - Actions and Outlets
iOS - Delegates
iOS - UI Elements
iOS - Accelerometer
iOS - Universal Applications
iOS - Camera Management
iOS - Location Handling
iOS - SQLite Database
iOS - Sending Email
iOS - Audio & Video
iOS - File Handling
iOS - Accessing Maps
iOS - In-App Purchase
iOS - iAd Integration
iOS - GameKit
iOS - Storyboards
iOS - Auto Layouts
iOS - Twitter & Facebook
iOS - Memory Management
iOS - Application Debugging
iOS - Video Lectures
iOS - Exams And Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

IPHONE BASICS COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is iPhone?

iPhone is a smartphone that is designed, produced and distributed by Apple Incorporation that joins together the features of a computer, an iPod, a digital camera and a cellular phone into one device which comes with a touchscreen interface.

iPhone runs on the Apple iOS Operating System (OS), and as of 2017 there were approximately 2.2 million mobile applications that are available for it through the Apple App Store, according to a statistic.

Steve Jobs who was the former Apple CEO introduced the iPhone to the public on January 9, 2007, at the Macworld 2007 conference in San Francisco. It came up for sale on the 29th of June that same year.

Apple sold over 13 million iPhones in the first five quarters of the device launch to the market. In 2010, the number of iPhones sold since its release has gone over 70 million. In 2011, Apple announced that it had sold off 100 million iPhones. That same year, Apple overshadowed Nokia as the mobile phone manufacturer with the highest annual revenue.

The First-generation iPhone was extremely thin, only about 11.6 mm thick but it as much wider and longer than many like devices. The display area of the iPhone was a 3.5 inch-wide screen display with a multitouch interface and an unusually high resolution (at about 160 pixels per inch). Unlike most other smartphones that were released at that time, the iPhone did not use any hardware keyboard or a stylus pen. Instead, its user relied on multiple taps and drags to go through the iPhone OS 1.0, which is a mobile version of Apple's OS X for Mac computers. Like the iPod, the iPhone synchronized its user's data with a user's personal computer through a USB port, using the Apple iTunes as client software.

Features of an iPhone

There are lots of features of an iPhone and some of them are:

1. Email: Like all good smartphones out there, the iPhone has very robust email features and it can sync to corporate email servers that are running Exchange.

2. Calendar/PDA: The iPhone can be used as a personal information manager, with a very nice calendar application, address book, stock-tracking application, weather update, and other related features.

3. iPod: A shortcut description of an iPhone is that it is a cell phone and an iPod combined together, so of course its music player features offer all the awesome advantages and coolness of iPods.

4. Video playback: With its big, and beautiful, the iPhone is a great choice for mobile video playback, whether you are using the built-in YouTube application, adding your own video, or buying or renting video contents from Apple iTunes Store.

5. Apps: With the addition of the Apple App Store, iPhone users can now install all kinds of third-party application programs, ranging from games (both free and paid versions), Facebook and Twitter, restaurant finders up to its long list of productivity apps. The App Store makes the iPhone stand as the most useful smartphone around.

6. Cameras: One major change in the iPhone is the inclusion of more cameras, whereas previous models only had one. The user-facing camera of the iPhone allows for FaceTime video chats.

Benefits of an iPhone

There are many benefits of the iPhone and some of them are:

1. User-Friendly: iPhone carves a good reputation for itself by being the most user-friendly phone to date. The most interesting feature about the iPhone is the look and texture of iOS, till today it's look hasn't changed, It is exactly the way it was launched in 2008. iPhone is also easy to learn and navigate.

2. Better Integration Of Hardware And Software: It is crucial for a phone to have the correct integration of hardware and software. iPhone 7 has a touch display that can detect the pressure which is being put on it. Another important benefit of using an iPhone is its sensitivity level adjustment.

3. No Bloatware: No bloatware issues with an iPhone, although apps come preloaded in iPhone that may not be needed, still you can upload a lot of unwanted stuff.
4. Fast: iPhones are faster than Android phones, it's always found among top fast phones on the media. It's also smooth, call quality, facetime feature, with background apps open one browsing.

5. Awesome applications: iPhone has the best apps by far, apps which could navigate video call would normally be downloaded in Android, but can easily be done with an iPhone. The intriguing part is all apps available in the iPhone store are verified and completely safe to use.

6. Data Synchronization: Mac works perfectly with iPhones which are a very amazing combination. The continuity feature in macOS allows you to send and receive messages over MacBook along with calls, so long as both are near.

7. Apple Pay: So easy to use, by just navigating an iPhone to a payment terminal with your finger on the Touch ID, you can make payments online.
8. Family Sharing: This allows families to share apps, images, etc. from their iBooks, App Store and iTunes easily, still maintaining personal iTunes, and access to a 200G cloud storage for the family.

Why Study iPhone Basics

1. Understand the workings of iPhone device and OS as a developer, analyst or IT professional.

2. Gain knowledge on the speed of the OS and the applications that runs on it.

3. Understand the awesome productivity and utility applications.

4. Gain knowledge on security and data synchronization features.

5. Increase your earning potential.

6. Job opportunities and career advancement.

iPhone Basics Course Outline

iPhone Basics - Buying an iPhone
iPhone Basics - Getting Started with the iPhone
iPhone Basics - Using iOS
iPhone Basics - Using the Keyboard
iPhone Basics - Using the Phone App
iPhone Basics - Adding and Managing Contacts
iPhone Basics - Browsing with Safari
iPhone Basics - Installing and Managing Apps
iPhone Basics - iPhone Settings
iPhone Basics - Security and General Settings
iPhone Basics - Wallpaper and Sounds
iPhone Basics - Managing Notifications
iPhone Basics - Syncing Your iPhone
iPhone Basics - Apple Apps
iPhone Basics - Using The Mail App.
iPhone Basics - Messages
iPhone Basics - Calendar
iPhone Basics - Reminders
iPhone Basics - Camera and Photos
iPhone Basics - iPhone Tips and Tricks
iPhone Basics - Hidden iOS Features
iPhone Basics - Video Lectures
iPhone Basics - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

IPAD BASICS COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is an iPad?

iPads are smart mini computer devices designed, produced and distributed by Apple Incorporation that joins together the features of a computer, an iPod, a digital camera and a cellular phone into one device which comes with a touchscreen interface.

iPads run on the Apple iOS Operating System (OS), and as of 2017 there were approximately 2.2 million mobile applications that are available for it through the Apple App Store, according to a statistic.

Features of an iPad

There are lots of features of an iPad and some of them are:

1. Email: Like all good smartphones out there, the iPad has very robust email features and it can sync to corporate email servers that are running Exchange.

2. Calendar/PDA: The iPad can be used as a personal information manager, with a very nice calendar application, address book, stock-tracking application, weather update, and other related features.

3. iPod: A shortcut description of an iPad is that it is a cell phone and an iPod combined together, so of course its music player features offer all the awesome advantages and coolness of iPods.

4. Video playback: With its big, and beautiful, the iPad is a great choice for mobile video playback, whether you are using the built-in YouTube application, adding your own video, or buying or renting video contents from Apple iTunes Store.

5. Apps: With the addition of the Apple App Store, iPad users can now install all kinds of third-party application programs, ranging from games (both free and paid versions), Facebook and Twitter, restaurant finders up to its long list of productivity apps.

6. Cameras: One major change in the iPad is the inclusion of more cameras, whereas previous models only had one. The user-facing camera of the iPad allows for FaceTime video chats.

Benefits of an iPad

There are many benefits of the iPad and some of them are:

1. User-Friendly: iPad carves a good reputation for itself by being the most user-friendly phone to date. The most interesting feature about the iPad is the look and texture of iOS, till today it's look hasn't changed.

2. Data Synchronization: Mac works perfectly with iPad's which are a very amazing combination. The continuity feature in macOS allows you to send and receive messages over MacBook along with calls, so long as both are near.

3. Apple Pay: So easy to use, by just navigating an iPad to a payment terminal with your finger on the Touch ID, you can make payments online.

4. Family Sharing: This allows families to share apps, images, etc. from their iBooks, App Store and iTunes easily, still maintaining personal iTunes, and access to a 200G cloud storage for the family.

Why Study iPad Basics

1. Understand the basic and fundamental functions of an iPad.
2. Suitable for professionals who are always on the go.
3. Increase your knowledge on iPad operations and design architecture.
4. Increase your earning potential as a mobile publisher.
5. Job opportunities and career advancement.

iPad Basics Course Outline

iPad Basics - Introduction
iPad Basics - Buying an iPad
iPad Basics - Getting Started with the iPad
iPad Basics - Using iOS
iPad Basics - Using the Keyboard
iPad Basics - Browsing with Safari
iPad Basics - Installing and Managing Apps
iPad Basics - iPad Settings
iPad Basics - Security and General Settings
iPad Basics - Wallpaper and Sounds
iPad Basics - Managing Notifications
iPad Basics - Syncing Your iPad
iPad Basics - Mail
iPad Basics - Contacts
iPad Basics - Calendar
iPad Basics - Reminders
iPad Basics - Messages
iPad Basics - Camera and Photos
iPad Basics - Extras
iPad Basics - Things You Didn't Know You Could Do with Your Smartphone
iPad Basics - Hidden iOS Features
iPad Basics - Video Lectures
iPad Basics - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

JAVA PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is JAVA?

JAVA is a high-level, object-oriented programming language that was initially created by Sun Microsystems and published in 1995. Java programs can be run on a lot of platforms, such as Windows, Linux, and Mac.

JAVA is widely known to be multi-purpose programming language that is designed to be both concurrent, class-based, object-oriented, and also it is specifically programmed to run with very few dependencies as possible in any machine. It is intended to let software application developers **"write once, run anywhere"** hence the word **WORA,** meaning that Java codes after being compiled, can run on all the platforms that support Java without the need for you to recompile the code.

For example, you can code and compile a Java program on a Linux machine and run it on Microsoft Windows, Mac, or another Linux machine without any change to the source code. **WORA** is made possible by compiling a Java program into an intermediate language known as bytecode. The format of java is platform-independent. A virtual machine, known as the **Java Virtual Machine (JVM),** is used to run the bytecode on various platforms.

Features Of JAVA Programming

There Are Lots Of Features Of Java And Some of Them Are:

1. Object-Oriented: everything in Java Logic is an Object. Java can be easily modified and extended since it is based on the Object model.

2. Platform Independent: Unlike many other programming languages, for example, C and C++, when a Java Code is compiled, it is not directly compiled into machine code, instead it is compiled into a platform-independent byte code. This byte code can be distributed over the internet and can then be interpreted by the **Java Virtual Machine** (JVM) on whatever your platform of choice is that it would be run on.

3. Simple: Java is created with a goal to make it easy to learn. If you can understand how the Object-Oriented design concept works in java, it would be easy to master Java.

4. Secure: With Java's security features, implemented from the start of development, you can develop tamper-free systems. Also, authentication techniques are solely dependant on public-key encryption.

5. Architecture Neutral: Java compiler produces an object file format that is architecturally neutral, and it also makes it possible for the compiled code to be executable on many processors, with the presence of a runtime system.

6. Portable: Being architecture-neutral and having no software implementation-dependent aspects of the specification makes Java programs to be portable. Java compilers are written in ANSI C with a boundary on clean portability, which is a subset of POSIX.

7. Robust: Java makes an effort to remove errors by making error checking very strict and also checking for error at runtime.

Benefits Of JAVA Programming

1. Multithreaded: With Java's multithreading feature you can now write programs that can carry out many tasks simultaneously. This design feature makes it possible for the developer to create highly interactive applications that can run very smoothly.

2. Interpreted: Java byte code is translated on the go into native machine instructions and is not saved anywhere else. The process of development with Java is more fast and analytical since it is a step by step and light-weight process.

3. High Performance: With the use of Just-In-Time compilers, Java makes sure that your programs are developed with very high performance.

4. Distributed: Java is created for the distributed environment of the internet.

5. Dynamic: Java is seen by many as a language that is more dynamic than C or C++ because Java is designed to easily adapt to a changing environment. Java programs can carry a number of run-time information that can be used to verify and resolve objects quickly.

This extensive course would give you a complete understanding of JAVA Development.

JAVA Programming Course Outline

Java - Introduction/Overview
Java - Environment Setup
Java - Basic Syntax
Java - Object & Classes
Java - Constructors
Java - Basic Datatypes
Java - Variable Types
Java - Modifier Types
Java - Basic Operators
Java - Loop Control
Java - Decision Making
Java - Numbers
Java - Characters
Java - Strings
Java - Arrays
Java - Date & Time
Java - Regular Expressions
Java - Methods
Java - Files and I/O
Java - Exceptions
Java - Inner classes
Java - Inheritance
Java - Overriding
Java - Polymorphism
Java - Abstraction
Java - Encapsulation
Java - Interfaces
Java - Packages
Java - Data Structures
Java - Collections
Java - Generics
Java - Serialization
Java - Networking
Java - Sending Email
Java - Multithreading
Java - Applet Basics
Java - Documentation
Java - Video Lectures
Java - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

JULIA PROGRAMMING LANGUAGE COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Julia Programming?

Julia Programming Language is a high-performance, high-level, dynamic computer programming language. While it is an all-purpose language and can be used to write any software application, most of its features are very well suited for numerical analysis and computational science.

The distinctive aspects of Julia's design include a data type system that has parametric polymorphism in a dynamic programming language; with several dispatch as its core programming paradigm. Julia offers support for parallel, concurrent, and distributed computing and for the direct calling of C and Fortran libraries without any glue code.

Julia is a garbage-collected language, makes use of eager evaluation, and it consists of efficient libraries for floating-point calculations, random number generation, linear algebra, and regular expression matching. Several libraries are available, which includes some that were formerly packaged with Julia and are now made separate.

Features and Benefits of Julia Programming Language

Julia is really good at carrying out scientific computing, but not restricted to doing just that, as it can also be used for web and all purpose programming tasks. Some of Julia's features are mentioned as follows:

1. Julia is an expressive, modern, high-performance computer programming language that was designed for scientific computation and data manipulation.

2. It is designed and distributed for parallel computation.

3. Julia offers a well extensive library of mathematical functions that has great numerical accuracy.

4. While the Julia's ecosystem is not as mature as C++, R or Python , the growth rate of the penetration of the language is doubling by the day.

5. The Pycall package gives Julia the ability to call Python functions from within its code and MATLAB packages using the MATLAB.jl package. Functions and libraries that are developed in C can also be called directly without any need for new APIs or wrappers.

6. Julia provides a powerful shell-like capabilities for managing other processes that are in the system.

7. Unlike other languages, user-defined types in Julia are well compact and are quite fast as built-ins.

8. Scientific computations makes great use of vectorized code to gain more performance benefits. Julia eliminates the need to vectorize code to attain any performance. Devectorized codes that are written in Julia can be as fast as the vectorized codes.

9. Julia has an elegant and well extensible conversions and promotions for both numeric and other types.

10. Julia is a free and open source language with an MIT license.

Why Study Julia Programming Language

There are many benefits of the studying Julia Programming language, and some of them are:

1. Gain extensive development skills with Julia.

2. Julia makes use of cooperative multitasking, lightweight green threading also refered to as tasks or coroutines, or one-shot continuations.

3. Julia has a very powerful type system. The conversions Julia provided by Julia Type systems are elegant and well extensible.

4. Julia has well efficient support for Unicode

5. Julia has several facilities for metaprogramming and for Lisp-like macros.

6. Julia has a built-in package manager known as Pkg.

7. Julia offers job opportunities and career advancement.

8. Enrich your CV and increase your earning potential.

Julia Programming Course Outline

Julia Programming - Getting Started
Julia Programming - Variables
Julia Programming - Integers and Floating-Point Numbers
Julia Programming - Mathematical Operations and Elementary Functions
Julia Programming - Complex and Rational Numbers
Julia Programming - Strings
Julia Programming - Functions
Julia Programming - Control Flow
Julia Programming - Scope of Variables
Julia Programming - Types
Julia Programming - Methods
Julia Programming - Constructors
Julia Programming - Conversion and Promotion
Julia Programming - Interfaces
Julia Programming - Modules
Julia Programming - Documentation
Julia Programming - Metaprogramming
Julia Programming - Multi-dimensional Arrays
Julia Programming - Missing Values
Julia Programming - Networking and Streams
Julia Programming - Parallel Computing
Julia Programming - Asynchronous Programming
Julia Programming - Multi-Threading
Julia Programming - Multi-Processing and Distributed Computing
Julia Programming - Running External Programs
Julia Programming - Calling C and Fortran Code
Julia Programming - Handling Operating System Variation
Julia Programming - Environment Variables
Julia Programming - Embedding Julia
Julia Programming - Code Loading
Julia Programming - Profiling
Julia Programming - Stack Traces
Julia Programming - Performance Tips
Julia Programming - Workflow
Julia Programming - Style Guide
Julia Programming - Unicode Input
Julia Programming - Base
Julia Programming - Standard Library
Julia Programming - Developers Documentation
Julia Programming - Video Lectures
Julia Programming - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

K PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is K Programming?

K Programming is a proprietary array processing software programming language that was developed by Arthur Whitney and later made a commercial product by KX Systems.

K Programming Language serves as the base for kdb+, which is a column-based, in-memory, database, and other related financial products. The language which was originally developed in 1993, is a modification of APL and is made up of elements of Scheme. Advocates of the language maintain its facility/speed in handling arrays, and expressive syntax, and is a real-world language made for real use and sold as a real product, it is not intentionally obfuscated or made to be intentionally strange.

K Programming has the same key features as APL. They are both interpreted, object-oriented and interactive languages noted for their very expressive and concise syntax. They have manageable rules of precedence based on evaluation fro right to left. The languages are made up of a rich set of primitive functions that are designed for processing arrays. These primitive functions include mathematical operations that work on arrays as whole data objects, and array operations, such as reversing or sorting the order of an array. In addition, the language includes special operators that merge with primitive functions to carry out various types of iteration and recursion. As a result, complex and extended dataset transformations can be represented as a chain of sub-expressions, with each link handling a segment of the calculation and passing the results to the next link in the chain.

Like APL, the primitive functions and operators are designated by single or double characters; however, unlike APL, K limits itself to the ASCII character set. To allow for this, the collection of primitive functions in K is smaller and heavily overloaded, with each of the ASCII characters depicting two or more distinct functions or operations. In a given expression, the actual function that is referenced is determined by the context. As a result, expressions in K can be opaque and difficult to parse for humans.

Features of K Programming

Some of the features of K include:

1. Object-Oriented: K is completely an object-oriented programming language. Each and every value in K is an object. Every object has a class and every class has a superclass. Every single code has its characteristics and actions.

2. Flexibility: K is a flexible language as you can easily redefine, remove or add existing parts to it. It allows its developers to freely modify its parts as they wish.

3. Mixins: K has a feature of single inheritance only. K has modules as well as classes. A module has methods but it has no instances. Rather, a module can be mixed into a class, which then combines the method of that module to the class. For each satisfiable class, provide an example of an instance with satisfying assignments to the properties. Note, this is not the only possible solution to the class, rather one of the potentially many solutions. If no solution is possible (i.e. the class is NOT satisfiable), then K will provide you a corresponding report.

4. Visual Appearance: K programming generally prefers keyword in English.

5. Dynamic Typing: K is a dynamic software programming language. Programs developed with K are not compiled. All modules, classes and method definitions are completed by the code when it executes.

6. Variable Constants: In K, constants are not actually constant. If a constant that has already been initialized is to be modified in a script, it will simply bring up a warning but will not stop your program from executing. It proves that classes with given constraints are satisfiable. If your constraints are specified at the top level without any classes, that is fine as well. The constraints will be proven satisfiable at the top level.

7. Primitive Types: K currently has the following primitive types: Int, Real, Char, String, Bool.

8. Classes: Classes are a basic method of encapsulating all relevant information, properties, functions, etc. about a single entity. They provide the simplest approach for abstracting a concept or concrete entity and representing it in your model/program.

9. Properties: Properties are attributes that you want to be associated with a particular concept.

10. Inheritance: K supports inheritance (multiple as well as single) between classes. Inheritance works in a manner similar to other languages such as Java, with the one difference that in K, sub-classes not only inherit the properties and functions of the parent class(es) but also the constraints.

11. Functions: K supports the definition of functions, which essentially are named parameterized expressions. A function can be applied to a list of arguments and return a value.

Benefits of K Programming

There are many benefits and advantages of K programming and some of them are:

1. K is an open-source and community development project.

2. The learning curve of K is very swift, newbies to programming can pick up with the syntax easily.

3. K has a user-friendly data Structure.

4. Productivity and Speed with K are very high.

5. K has extensive support for third-party libraries and modules.

6. K is an object-oriented software programming language

7. Exceptional language for dealing with mathematical analysis, financial prediction and handling bulk data.

8. K has bindings to other languages such as Java, C, Visual Basic, and Excel.

9. Though an interpreted language the source code is compiled internally

10. It's built-in interprocess communication and binary format for objects is very simple and documented, making other system interaction easy.

Why Study K Programming

1. K treats everything as an object, meaning that you can use it to model real-world applications.

2. K can be used to develop systems applications and software that are highly scalable.

3. Since K compiles its source code internally, it is extremely fast.

4. Job opportunities and career advancement.

K Programming Course Outline

K Programming - Introduction
K Programming - Architecture
K Programming - Overview
K Programming - Architecture
K Programming - Syntax
K Programming - Advanced Topics
K Programming - Attributes
K Programming - Functional Queries
K Programming - Advanced Topics
K Programming - Table Arithmetic
K Programming - Tables on Disk
K Programming - Maintenance Functions
K Programming - Video Lectures
K Programming - Exams And Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

MATLAB COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What Is MATLAB?

MATLAB which is the acronym for **Matrix Laboratory** is a software programming language that is developed by MathWorks.

MATLAB started out initially as a **Matrix Programming Language** which aim is to make linear algebra programming very simple. MATLAB programs can be run both under interactive sessions and also as a batch job.

MATLAB is a fourth-generation high-level programming language and interactive environment that is mainly used for numerical computation, data visualization, and programming.

MATLAB allows you to carry out matrix manipulations, plotting of functions and data, carrying out the implementations of algorithms; creating user interfaces, accessing programs that are written in other languages, including C, C++, Python, Java, and Fortran. It also allows you to analyze any given data, produce algorithms to solve the problems and create models of the software and applications.

MATLAB has lots of in-built commands and mathematical functions that allow you to carry out mathematical calculations, generate plots, and to perform numerical methods. MATLAB is an interactive development system whose basic data element is a list (array) that does not need or depends on dimensioning, this allows you to solve many technically related problems in computing, especially those that involve the formulations of matrix and vector in just a fraction of the time that it would take to write the program in a scalar and non-interactive language such as C++ or Java.

Features Of MATLAB

The MATLAB Programming language has a lot of features included in it which is due to it being a data visualization language, the following are some of the basic features of MATLAB.

1. MATLAB is a high-level language that is used for numerical computation, data visualization, and software application development.

2. MATLAB also gives you an interactive development environment for iterative exploration, program design and problem-solving.

3. MATLAB also comes with a vast library of mathematical functions for solving linear algebra, Fourier analysis, statistics, data filtering, optimization, numerical integration and for solving regular differential equations.

4. MATLAB provides you with built-in graphics tools for visualizing data and it also gives you tools for creating custom plots as per your application design.

5. The MATLAB's programming interface comes with lots of development tools that help you to improve the quality of your code and code maintainability and maximizing program performance.

6. MATLAB also provides you with tools for building software applications that have custom graphical user interfaces.

7. Lastly MATLAB provides functions for using algorithms that are based on MATLAB on external applications and languages such as Java, C++ .NET language and Excel.

Benefits Of MATLAB

There are lots of benefits and advantages of learning the MATLAB programming language and some of them are:

1. MATLAB is a tool that is independent of the compiler to be used and is doesn't require any compiler to run unlike languages like C, C ++ where it is required. Codes in MATLAB are written in sentences and executed one after the other.

2. Being compiler independent makes Matlab more productive and efficient.
3. MATLAB is a fourth-generation high-level programming language.

4. MATLAB coder is used to convert the code that is written in MATLAB to languages like Python, JAVA, C++, .NET, etc thereby making the MATLAB language more versatile.

5. Different programming languages can be used to perform scientific theories and after building the library files or .dll files they can be directly executed in Matlab using any other languages.

6. Data in MATLAB can also be stored in variables with very simple commands that are easy to use by storing numbers in a vector or matrix where the use of loops is not needed.

7. Complex mathematical operations such as the multiplication and addition of matrix can be easily achieved in a single code by using MATLAB.

MATLAB Study

This MATLAB programming language course is designed to give students a very clear understanding of MATLAB programming language

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

Node JS COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What Is Node JS?

Node JS is a server-side platform that is built on Google Chrome's JavaScript Engine (V8 Engine). The Node JS library was developed by Ryan Dahl in 2009 and its current version as of the time of this writing is v 13.0.1.

Node.js is a javascript library that is built on Google Chrome's JavaScript runtime engine to enable software developers to easily build very fast and scalable network applications. Node.js makes use of an event-driven, non-blocking Input-Output model that makes it lightweight, very effective and perfect for real-time web applications that highly depend on data that run across multiple devices.

Node.js is an Open-Source, multi-platform development environment that is used for developing server-side and Networking Applications. Node.js applications are generally written in JavaScript and they can be run inside the Node.js runtime on Microsoft Windows, OS X and the flavors of Linux.

The Node.js library also offers a rich set of various **JavaScript Libraries and Modules** that makes easy, the process of developing easily scalable web applications using Node.js to a great extent.

Features Of Node JS

Node JS has so many features and characteristics. The following are some of the important features that make Node.js to become the first choice of software developers.

1. Asynchronous and Event-Driven: All APIs found in the Node.js library are asynchronous, which means that they are non-blocking. It basically means that a web server that is based or built upon Node JS does not wait for an API to return request data. The server goes straight to the next API after requesting the API resource and a notification tool of Events in Node.js helps the web-server to get a response from the API that was called previously.

2. Very Fast: Node JS library, Being built on Google Chrome's V8 JavaScript Engine, is very fast in executing the code that is being run on it.

3. Single-Threaded but Highly Scalable: Node.js makes use of a single-threaded model that includes looping of events. The event mechanism of Node JS helps the server to respond in a non-blocking way and it makes the server highly scalable which is opposing to traditional web servers that use the threading feature to create limited threads that handle requests. Node.js web servers use a single-threaded program and the same single-threaded program can provide better function and service to a much larger number of requests than conventional web servers like the Apache HTTP Server.

4. No Buffering: Node.js applications does not buffer any data. Its applications simply output the data in pieces.

5. License: The Node JS library is released under the MIT license.

Why Study Node JS

Any computer programming language will give you some reasons to prefer them over the others. The thing about Node.js is that it was designed and developed with the sole purpose of managing applications at scale. Modern tooling and its embrace of the more presumptuous-looking way of building complex web applications make Node.js stand out from other computer programming technologies.

With over a billion downloads, Node.js shines in developing real-time software applications, internet of things applications, and several microservice applications. It is gaining traction faster than any other software technology and it ranks in the top most in-demand software developer skills.

Below are some of the reasons why you should study Node JS:

1. Performance and Scalability: Node.js is built on top the Google Chrome's V8 runtime engine, it is developed with C++, designed for several software operating systems and it is extremely fast. Both chrome V8 and Node are updated frequently, with several optimizations for performance, security patches, and its support for modern JavaScript features. The use of JavaScript also signifies that converting JSON data which is the most common data interchange format on the Web is fast by default.

2. Data Streaming: In more conventional web platforms, HTTP requests and responses are being treated as an isolated event; in fact, they actually stream of data. This observation can be employed in Node JS to develop some cool features. For example, it is very possible to process files while they're still being uploaded to the server, as the data comes into the server through a stream and we can process it in an online manner. This could be carried out for real-time encoding of audio or video files.

3. Proxy: Node.js is easily used as a server-side proxy where it can manage a large number of concurrent and simultaneous connections in a manner that is non-blocking. It's particularly useful for proxying various services with different response times or gathering data from several source points.

4. Queed Inputs: If you're getting a high amount of simultaneous and concurrent data, your database can become a bottleneck. Node.js can easily manage the concurrent connections themselves.

Career Opportunities in Node JS

At the entry/ beginner level, most people start their career as a front-end developer who develops software applications using HTML, CSS, and JavaScript. Then to make things more convenient, and powerful, additional technologies need to be included in the project and Node JS is one such technology.

After spending a few years as a developer with this technology candidates are normally promoted as a team leader, architect, scrum master and product owner. Those who prefer to move towards the management-based roles usually become team manager, people manager, or line manager.

Many developers also have the option to own their businesses or start-ups. Such as web development consulting, online and classroom-based tutorials, book writing and teaching. Game development, app development and product development can also be an exciting career option for Node.js professionals and freelancers.

Benefits And Advantages Of Node JS

1. Node.js offers an easy way to scale your application that is built on it.
2. Node JS is very easy to Learn
3. Node.js is used as a Single Programming Language for JavaScript.
4. Node JS gives you the benefit of being a Fullstack JS developer.
5. Node JS is known for Offering High-Performance I/O operations.
6. Node JS developers have the Support of a Large and Active developer Community.
7. Node JS has The Advantages of Caching
8. Node JS offers you The freedom to Develop Apps using JS.
9. Node JS gives you support for commonly used development tools.
10. Node JS handles API Requests Simultaneously.
11. Node.js is Highly Extensible.

Node JS Course Outline

Node.js - Introduction
Node.js - Environment Setup
Node.js - First Application
Node.js - REPL Terminal
Node.js - Package Manager (NPM)
Node.js - Callbacks Concept
Node.js - Event Loop
Node.js - Event Emitter
Node.js - Buffers
Node.js - Streams
Node.js - File System
Node.js - Global Objects
Node.js - Utility Modules
Node.js - Web Module
Node.js - Express Framework
Node.js - RESTFul API
Node.js - Scaling Application
Node.js - Packaging
Node.js - Video Lectures
Node.js - Exams And Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

PASCAL PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Pascal?

Pascal Programming Language is an imperative, procedural and object-oriented programming language that was developed by Niklaus Wirth as a small, but very efficient programming language, designed to encourage good software development practices by employing both structured programming and data structuring. It was given the name Pascal in honor of the French philosopher, mathematician, and physicist Blaise Pascal.

Based on the book that was written by Wirth's, Algorithms + Data Structures = Programs, Pascal was built on the pattern of the ALGOL 60 language. Wirth was actively involved in the process to better the language as part of the ALGOL X efforts and brought up a version that was referred to as ALGOL W. This version was not accepted, and the ALGOL X process took it down. In 1968, Wirth made up his mind to abandon the ALGOL X process and further improve ALGOL W, therefore releasing this to the general public as Pascal in 1970.

On top of the scalars and arrays of ALGOL's, Pascal allows developers to define complex datatypes and to develop dynamic and recursive data structures such as lists, graphs, and trees. Pascal has a strong data type on all its objects, which means that one form of data cannot be modified or interpreted as another without explicitly converting it.

Unlike other programming languages in the C-family, Pascal allows you to use a nested definition of procedures to any level of depth, and it also allows almost any kind of data definitions and declarations inside its subroutines (functions and procedures). A Pascal program is therefore similar syntactically to a single procedure or function.

Pascal became very popular and successful in the 1970s, mainly on the burgeoning minicomputer market. Various compilers were also made available for many microcomputers as the field came up in the late 1970s. It was broadly used as a language for teaching in university-level computer programming courses in the 1980s, and it is also used in production areas for developing commercial software during the same period. It was made almost obsolete by the C programming language in the late 1980s and early 1990s as Linux and UNIX-based systems became very popular, especially with the release of C++.

A derivative referred to as Object Pascal was developed for object-oriented programming in 1985, the Apple Computer and Borland made use of this in the late 1980s that was later developed into Delphi on the Microsoft Windows platform. Various extensions of the Pascal concepts brought about the languages Modula-2 and Oberon.

Features of Pascal

Pascal has the following features
1. Pascal is a strongly typed objective-oriented and procedural programming language.
2. Pascal offers an extensive error checking feature.
3. Pascal offers various data types like records, arrays, files, and datasets.
4. Pascal offers a list of software programming data structures.
5. Pascal allows for structured programming concepts through the use of functions and procedures.

Benefits of Pascal

There are many benefits of Pascal, and some of them are:
1. Pascal is a very clean programming language
2. Pascal has no Makefiles
3. Pascal compilers are very Fast.
4. Each unit in Pascal has it's own unique identifiers
5. Pascal programs are very fast, and they consume a lesser amount of memory.

Facts About Pascal

1. The Pascal language was named for Blaise Pascal, French mathematician, and pioneer in computer development.

2. Niklaus Wirth completed the development of the original Pascal programming language in 1970.

3. Pascal is based on the block-structured style of the Algol programming language.

4. Pascal was created as a suitable language for teaching programming as a systematic discipline, whose implementations could be both reliable and efficient.

5. The ISO 7185 Pascal Standard was originally published in 1983.

6. Pascal was the primary high-level language used for development in the Apple Lisa, and in the early years of the Mac.

7. In the year 1986, Apple Computer unveiled the very first object Pascal design, and also in the year 1993, the Pascal Standards Committee published an Object-Oriented Extension to Pascal.

Compilers and Interpreters for Pascal

There are various Pascal compilers and interpreters that are available for use. Among them are –

1. Turbo Pascal: provides an IDE and compiler for running Pascal programs on CP/M, CP/M-86, DOS, Windows, and Macintosh.

2. Delphi: This provides compilers for running Object Pascal and generates native code for 32- and 64-bit Windows operating systems, as well as 32-bit Mac OS X and iOS. Embarcadero is planning to build support for the Linux and Android operating systems.

3. Free Pascal: It is a free compiler for running Pascal and Object Pascal programs. Free Pascal compiler is a 32- and 64-bit Turbo Pascal and Delphi compatible Pascal compiler for Linux, Windows, OS/2, FreeBSD, Mac OS X, DOS, and several other platforms.

4. Turbo51: It is a free Pascal compiler for the 8051 families of microcontrollers, with Turbo Pascal 7 syntax.

5. Oxygene: It is an Object Pascal compiler for the .NET and Mono platforms.

6. GNU Pascal (GPC): It is a Pascal compiler composed of a front end to GNU Compiler Collection.

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

PYTHON PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Python?

Python Programming Language is a high level, object-oriented and multi-purpose programming language that comes with an integrated dynamic semantics which is basically for software and apps development. Python was created by Guido van Rossum and first released in the year 1991.

Python Programming is extremely attractive in the field of Rapid Application Development because it is dynamically typed and has dynamic binding options.

Python is also popular because it is relatively simple and easy to learn since it is made up of a unique syntax that enforces readability. **Software Developers** can read and understand Python codes much easier than they can with other Programming Languages. The major effect of this is that it brings down the cost of developing and maintaining programs because it allows development teams to work together without really noticing the language and experience difference between them.

One of the most interesting features of the Python Programming language is that both its standard library and the interpreter are available free of charge, in both binary and source form. There is no special difference either, as Python and all the necessary tools are available on all the major platforms available. Therefore, it is a nice and interesting option for developers who don't want to worry about paying a high price for developing software.

Python is a general-purpose programming language, which is another way for us to say that Python can be used for almost everything. Most importantly, Python is an interpreted language, which is to say, that the written code is not actually translated to machine level during runtime. Even though, most Programming Languages do this conversion before the program can even be run. This type of Programing language is also referred to as a "scripting language" because it was initially meant to be used for different smaller tasks.

Features Of Python Programming

Python comes with a lot of features out of the box, some of them are listed below:

1. Python is Easy to Learn and Use: Python is very easy to learn and use. It is a developer-friendly and high-level object-oriented programming language.

2. Expressive Language: Python language is more expressive which is to say that it is more understandable and readable.

3. Interpreted Language: Python is an interpreted language. For example, the interpreter executes the code one line at a time. This feature makes debugging python code easy and therefore suitable for beginners.

4. Cross-platform Language: Python can run the same way on different platforms such as Windows, Linux, Unix, and Macintosh, etc. We can, therefore, say that Python is a portable language.

5. Free and Open Source: Python language is freely distributed at its official web address. The source code is also available for everyone to see and contribute to its development. Therefore it is an open-source language.

6. Object-Oriented Language: Python supports an object-oriented language paradigm that brings the concepts of classes and objects to come into existence.

7. Large Standard Library: Python has a very large and broad standard and third-party libraries and it provides a rich set of modules and functions for rapid application development.

8. GUI Programming Support: Graphical user interface software can be developed using Python.

9. Integrated: Python can be easily integrated into other languages such as C, C++, JAVA, Javascript, etc.

Benefits Of The Python Programming Language

There are lots of benefits of the python programming language some of them are:

1. Python has a huge category of Third-Party Modules.
2. Python has an extensive Support Libraries.
3. Python has a large Open Source and Community Development.
4. The learning curve of python is easy and it has Support Available.
5. Python has user-friendly Data Structures.
6. Productivity and Speed.
7. It provides supports very functional and properly structured programming technique as well as OOP.
8. Python can be used mostly as a scripting language or can also be compiled to byte-code for building very large applications.
9. It offers quite a very high-level dynamic data types and also provides supports dynamic type checking.
10. It offers support for automatic garbage collection.
11. It can be easily combined with Java, COM, C++, COBRA, ActiveX, and C.

Career Opportunities in Python Programming

An increasing number of companies are bringing up the need for Python professionals. If you are in the look for career opportunities that are available in this field, they are large, and some of them are listed below.

1. You get a lot of job opportunities if you study Python.
2. You can become a Python Specialist in any organization.
3. After taking this certification course of the Python module, you can become a Software Developer, Backend Engineer or Machine Learning Engineer.
4. You can become a Software Engineer or a Python Engineering certified trainer.
5. You can become a Senior Python Engineer commanding high pay.

Python Course Outline

Python - Introduction/Overview
Python - Environment Setup
Python - Basic Syntax
Python - Variable Types
Python - Basic Operators
Python - Decision Making
Python - Loops
Python - Numbers
Python - Strings
Python - Lists
Python - Tuples
Python - Dictionary
Python - Date & Time
Python - Functions
Python - Modules
Python - Files I/O
Python - Exceptions
Python - Classes/Objects
Python - Reg Expressions
Python - CGI Programming
Python - Database Access
Python - Networking
Python - Sending Email
Python - Multithreading
Python - XML Processing
Python - GUI Programming
Python - Further Extensions
Python - Video Lectures
Python - Exams And Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

REACT NATIVE COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is React Native?

React Native is a JavaScript mobile framework for developing real and native mobile applications for both iOS and Android devices. It's based on React, Facebook's JavaScript library for building and developing user interfaces, but instead of targeting the browser, this time, it is used to target mobile platforms.

Web Developers can now develop mobile applications that will have the look and feel of native applications, all from the comfort of a JavaScript library that they already know and use regularly. An added benefit here is that most of the code you write here can be shared between various platforms, React Native makes it very easy to simultaneously develop applications for both Android and IOS from the same code base.

React Native applications, similarly to React for the web, are written using a mixture of JavaScript and XML-like markup, popularly known as JSX. Then, under the hood, the React Native summons the native rendering APIs in Objective-C (for iOS apps) or Java (for Android applications). Therefore, your application will run using real mobile UI components, not just web views, and will look and feel like any other mobile application out there. React Native also exposes the JavaScript interfaces for platform-specific APIs, so your React Native applications can directly access platform features such as the phone camera, microphone or the user's current location.

React Native presently supports both iOS and Android devices and has the potential ability to expand to future platforms as well soon.

Features and Benefits of React Native

1. Programming Language: The programming language of any framework is one significant area that greatly influences the developer's ability a lot. If a development framework makes use of a programming language that is popular and widely used, it constantly affects and influences the choice of a developer. React Native uses one such language, Javascript which currently is the most popular and widely used programming language. Javascript is one of the three fundamental technologies of the World Wide Web (WWW), the other two are HTML and CSS.

2. UI Focussed: React Native is totally focussed on UI design. It is very responsive and its rendering capabilities are by far the best out there.

3. Community: Talking about community, React Native's community is one of the greatest in the cross-platform mobile application development community. React.js development was started back in 2013 by the Facebook team and it was launched in 2015. Soon it was made an open-source software and its popularity greatly increased. Its popularity was further pushed forward by the developers contributing to it from all over the world leading to a very strong community presence.

4. Tried and Trusted: The React Native is one of the most trusted cross-platform application development frameworks. Built by the social media giant Facebook, it is tried enough to be trusted by all. Facebook uses React Native in all its mobile applications. This is enough for developers to trust React Native. Moreover, it used by many other companies such as Airbnb, Instagram, Skype, Walmart, Tesla, and Discord.

5. Development Time: The time spent while developing with React Native is considerably very short.

6. Support for Third-Party Libraries: The usage of third-party libraries is always a huge point in development. React Native gives you support for the huge support of third party libraries.

7. Community-Driven: React Native is fully community-driven.

8. For Re-Use and Cost Saving: React Native Allows for Maximum code reuse and cost-saving.

9. For Reload: React Native comes with both live reload and hot reload.

10. Performance: React Native provides strong performance for mobile environments.

11. Architecture: React Native has a modular and intuitive architecture similar to React.

Why Study React Native

The following are some of the reasons why you should learn React Native:

1. Allows Performing Development Faster, and, Therefore, It Costs Less: With React Native the same code of your application is used for deployment on both your Android and iOS platforms. React Native helps you to save time and money by reducing the development cycle and scaling the team that is involved in a software project. With React Native, it is very possible to reduce the efforts spent on development by almost 50% without reducing either quality or productivity.

2. Provides Cross-Platform: The cross Platform nature of React Native means that it's possible to develop an application for two platforms such as Android and iOS. The framework combines together all the advantages of mobile application development with the native React's environment power and agility.

3. The Efficiency of Native Apps Development Increases Many-Fold: React Native is completely on React ideas and, therefore, it allows you to develop strong mobile applications. The development of a native software application is regarded as less efficient and more productive. React Native reduces the time spent on the development cycle, and it makes it possible to develop and deliver software products in the fastest way. The development of mobile applications is adapted to the hybrid environment and it has native results.

4. React Native Is Part of the Leading Frameworks for Cross-Platform Mobile application development: React Native's popularity is continually growing. There is no need for you to learn Java, Objective-C or Swift for you to develop applications for Android and IOS respectively. There is no need to hire a team of software developers anymore. A Javascript developer with little ideas of native design patterns, and UI elements, for hybrid mobile app development, and APIs is well enough.

5. Transform Any Web Project into a Mobile Decision Easily: With React Native, a code is reusable. Only one update is required for development on two platforms. React Native clarifies and simplifies detecting bugs between codebases very well. React Native interface is very intuitive and modular. This means that developers who are not engaged in a project can easily understand it and continue its development.

6. React Native Won't Disappear In the Years to Come: React Native is adopted by a vast number of software developers. This is so because guys won't waste their time and efforts on a technology that can disappear in a couple of years. The proof is that the Facebook team has long-term plans to invest so much more in its growth and development.

7. Increase Your Earning Potential

8. Job Opportunities and Career Advancement

React Native Course Outline

React Native - Introduction
React Native - Overview
React Native - Environment Setup
React Native - App
React Native - State
React Native - Props
React Native - Styling
React Native - Flexbox
React Native - ListView
React Native - Text Input
React Native - ScrollView
React Native - Images
React Native - HTTP
React Native - Buttons
React Native - Animations
React Native - Debugging
React Native - Router
React Native - Running IOS
React Native - Running Android
React Native - View
React Native - WebView
React Native - Modal
React Native - ActivityIndicator
React Native - Picker
React Native - Status Bar
React Native - Switch
React Native - Text
React Native - Alert
React Native - Geolocation
React Native - AsyncStorage
React Native - Video Lectures
React Native - Exams And Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

R PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is R Programming?

R Programming is a programming language that is used for carrying out statistical and data computing analysis which is developed and supported by the R software foundation.

R Programming Language is widely used among data analysts, miners and statisticians for developing statistical software and data analysis tools. Data mining surveys, polls, and studies of scholarly literature databases show a large increase in popularity.

R Programming Language is one of the most popular languages if not the most popular that is used by data scientists, statisticians, to get, tidy up, study, visualize and display data.

Due to the expressive syntax and easy-to-use interface of R, its popularity has grown in recent times. A GNU package, source code for the R software development environment is written basically in C, Fortran, and in R itself and is available for free under the GNU General Public License. Binary versions that are pre-compiled are made available for various software operating systems. Although R has a robust command-line interface, there are several graphical user interfaces, such as RStudio, which is an integrated development environment.

R Programming Language is basically an implementation of the S programming language joined together with lexical scoping semantics, that is inspired by Scheme. The S programming language was created by John Chambers in 1976 while working at Bell Labs. R was developed by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is presently managed and developed by the R Development Core Team (of which Chambers is also a member). R was given its name, partly after the first names of the first two R authors and partly as a run on the name of S. The project idea was formed in 1992, with the release of its first version in 1995 and a stable beta version in 2000.

R Programming Language and all of its libraries were built with wide types of both statistical and graphical techniques and methods, including linear and nonlinear modeling, classical statistical tests, classification, clustering, among others. R is structured, modular, scalable and can be easily extended with the help of extensions and functions, and the R community is recognized for its very active contributions in the section of packages.

Most of R's default functions are written in R, this makes it quite easy for developers to know about the algorithmic choices that are made. For jobs that are computationally heavy, C, C++, and Fortran code can be linked together and ran at run time. More experienced users can write C, C++,.NET, Java, or Python code to play and manipulate objects in R directly. R is highly extensible by the use of user-developed packages for carrying out particular functions or in specific areas of study. Due to its origin from S, R has a much stronger object-oriented programming features than most statistical computing languages. Extending R is also made easy by its lexical scoping rules.

R is extensible by the means of functions and extensions, and the R community is noted and known for its regular contributions in terms of packages. Advanced users can write C, C++, Java, .NET or Python code to manipulate and use R objects directly. Due to its S heritage, R has stronger object-oriented programming facilities than most of these statistical computing languages. Extendng R is also easy because of its lexical scoping rules.

R Programming is also used to generate publication-quality graphs, and also mathematical symbols. Dynamic and interactive graphics are available by means of additional packages.

R also has Rd, its own LaTeX-like documentation style, which is used to produce comprehensive documentation, both online in a number of styles and in hard copy.

Features of R Programming

There are many features of R Programming that makes it the most used development platform for large scale statistical computing, some of them are:

1. R is a fully-developed, simple and effective software programming language that includes conditionals, loops, user-defined recursive functions with input and output facilities.

2. R has a very effective facility for data handling and storage.

3. R provides a suite of operators for calculations on arrays, lists, vectors, and matrices.

4. R provides a large, coherent and integrated collection of tools for data analysis.

5. R provides graphical facilities for data analysis and display either directly at the computer or printing on papers.

6. R runs on all platforms: The R software that you are writing on one particular platform can easily be converted to another platform without any issues. Cross-platform compatibility is an essential feature to have in the computing world of this generation, even Microsoft is making its software cross-platform like the .NET platform which is currently available on all platforms after realizing the benefits of technology that runs on all systems.

Benefits of R Programming Language

There are many benefits and advantages of R and some of them are:

1. R is free and Open Source meaning that you don't have to pay to use.

2. R has exemplary support for Data Wrangling.

3. R comes with an array of Packages and extensions.

4. R allows you to Plot and Graph trends.

5. R allows you to carry out machine learning operations.

6. R is very popular and its popularity is still increasing.

7. Learning and Understanding R will triple your chances of getting a job since it's one of the most sort after languages for Data Analytics and science.

8. R is being used by the biggest giants of the tech world so it's going to be around for a very long time.

Applications of R Programming

1. Data Science.

2. Statistical computing.

3. Machine Learning.

6. Health Care.

7. Manufacturing.

8. E-Commerce.

9. Finance.

10. Banking.

11. R is mainly used for descriptive statistics like central tendency, measurement of variability.

12. R is widely applied to do exploration data analysis.

13. R is also applied for analyzing both discrete and continuous probability distribution.

14. R is also used for building and developing statistical software packages and to carry out analytical processing in other software suites.

In the Full Course you will learn everything you need to know about **R Programming Language** with **Certificate** to showcase your knowledge and competence.

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

REXX PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Rexx Programming?

Rexx is the acronym for Restructured Extended Executor is an interpreted high-level software programming language that was developed at IBM by Mike Cowlishaw.

Rexx is a structured, high-level, multi-purpose programming language that was designed for ease of reading and learning. Proprietary and open-source Rexx interpreters are made available for a broad range of computing platforms; compilers exist for IBM mainframe computers.

Rexx is used as both a macro and scripting language, and is frequently used for data and text processing and for generating reports on data; these similarities of Rexx with Perl mean that Rexx applications work well in Common Gateway Interface (CGI) programming and it is really used for this purpose.

Rexx is the fundamental scripting language in some operating systems, and it is also used as an internal macro language in some other system software, such as SPFPC, THE, KEDIT, and the ZOC terminal emulator. Additionally, the Rexx programming language can be used for developing scripts and macros in any program that makes use of the Windows Scripting Host ActiveX scripting engine languages (for example, JScript and VBScript) if one of the Rexx engines is installed in the system.

Features of Rexx Programming

There are many features of Rexx Programming and some of them are:

1. Rexx has a very simple syntax.
2. It has the ability to route commands to multiple environments.
3. It has the ability to support functions, procedures, and commands associated with a specific invoking environment.
4. A built-in stack, with the ability to interoperate with the host stack if there is one.
5. Small instruction set containing just two dozen instructions.
6. Freeform syntax.
7. Case-insensitive tokens, including variable names.
8. Character string basis.
9. Dynamic data typing, no declarations.
10. No reserved keywords, except in the local context.
11. No include file facilities.
12. Arbitrary numerical precision.
13. Decimal arithmetic, floating-point.
14. A rich selection of built-in functions, especially string and word processing.
15. Automatic storage management.
16. Crash protection.
17. Content addressable data structures.
18. Associative arrays.
19. Straightforward access to system commands and facilities.
20. Simple error-handling, and built-in tracing and debugger.
21. Few artificial limitations.
22. Simplified I/O facilities.

Benefits of Studying Rexx Programming

There are many benefits and advantages to studying Rexx Programming, and some of them are:

1. Rexx programs are designed to be very readable and easy to understand since there is a minimum number of required special characters, punctuation or notations.

2. Rexx arithmetic is generally described as decimal arithmetic, including precision that is selected by the programmer rather than by the computer hardware.

3. The Rexx Programming sets no specific limits on the size of strings to be used (including those that are used to represent numbers). This takes away most of the headaches that often faced by programmers.

4. Rexx is a moderately small programming language. This makes it be approachable and easy to learn even for beginners. Even the Object-oriented Rexx programming that is currently under development only adds a very few new constructs to the core of the language.

5. The dynamic nature of the Rexx programming language makes it very suitable for interpretation. Interpreted languages allow you to carry out iterations very fast during development with very low time spent overhead per program written.

6. Rexx is a system-independent language. This presents developers with the advantages of portability and scalability of their applications.

7. Rexx has no globally reserved words. This ensures that robust computer programs can be written which will not be made invalid by future additions to the collection of language instructions. Not only does this mean that software programmers do not need to learn the keywords that they do not need to use, but this is also very important when they are only using the language for developing an extension for an application.

8. Job Opportunities and Career Advancement.

REXX Programming Course Outline

Rexx - Introduction
Rexx - Overview
Rexx - Environment
Rexx - Installation
Rexx - Installation of Plugin-Ins
Rexx - Basic Syntax
Rexx - Datatypes
Rexx - Variables
Rexx - Operators
Rexx - Arrays
Rexx - Loops
Rexx - Decision Making
Rexx - Numbers
Rexx - Strings
Rexx - Functions
Rexx - Stacks
Rexx - File I/O
Rexx - Functions For Files
Rexx - Subroutines
Rexx - Built-In Functions
Rexx - System Commands
Rexx - XML
Rexx - Regina
Rexx - Parsing
Rexx - Signals
Rexx - Debugging
Rexx - Error Handling
Rexx - Object-Oriented
Rexx - Portability
Rexx - Extended Functions
Rexx - Instructions
Rexx - Implementations
Rexx - Netrexx
Rexx - Brexx
Rexx - Databases
Rexx - Performance
Rexx - Best Programming Practices
Rexx - Graphical User Interface
Rexx - Reginald
Rexx - Web Programming
Rexx - Video Lectures
Rexx - Exams And Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

RUST PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Rust Programming Language?

Rust is a multi-paradigm programming language that is designed to optimize memory usage, safety and high performance. Rust is similar to C++ in the areas of syntax, but is designed to provide a much better memory safety while maintaining high performance.

Rust was originally developed by Graydon Hoare at the Mozilla Research institute, with several members like Dave Herman, Brendan Eich, and a host of others. The developers subsequently refined the language while developing the Servo layout or browser engine, and the Rust compiler. The compiler is a free and open-source software that comes with a dual-licensed under the MIT License and the Apache License 2.0.

Rust is designed to be a language for developing highly concurrent and highly safe computer systems, and maintaining boundaries that preserve very large-system integrity. This has brought about a feature set with much emphasis on safety, control of the memory layout, and concurrency. Rust is developed to be memory safe, and therefore, does not permit null pointers, dangling pointers, or data races in safe code. Data values can only be initialized by means of a fixed set of forms, all of which demand that their inputs be already initialized.

Rust does not make use of an automated garbage collection system like those that are used by Java, Go, or the .NET Framework. Instead, in rust, memory and other resources are managed, through the resource acquisition initialization (RAII) means/convention, with very optional counting of reference. Rust provides deterministic management of resources, with a very low overhead.

Features Of Rust Programming Language

There are many features of rust, and below are some of them:

1. Zero cost abstraction: In Rust, we can add abstractions without affecting the runtime performance of the code. It improves the code quality and readability of the code without any runtime performance cost.

2. Error messages: In C++ programming, there is an excellent improvement in error messages as compared to GCC. Rust goes one step further in case of clarity. Error messages are displayed with (formatting, colors) and also suggest misspellings in our program.

3. Type inference: Rust provides the feature of a Type inference which means that it determines the type of an expression automatically.

4. Move semantics: Rust provides this feature that allows a copy operation to be replaced by the move operation when a source object is a temporary object.

5. Threads without data races: A data race is a condition when two or more threads are accessing the same memory location. Rust brings about the feature of threads that are without data races because of the ownership system in Rust. The ownership system only transmits the owners of different objects to their different threads, and two threads can never have the same variable with write access.

6. Safe memory space allocation: In Rust, memory management is manual, i.e., the programmer has explicit control over where and when memory is allocated and deallocated.

Benefits Of The Rust Programming Language

1. Rust is memory safe. By design, Rust code can't have dangling pointers, buffer overflows or other types of memory-related errors.

2. Rust is as fast as C/C++ while it is far safer.

3. Rust is a general-purpose language that can be used for any purpose.

4. Rust is great at concurrent programming.

5. Rust has an inbuilt dependency and build management known as Cargo.

6. Rust has great communities.

Why Study Rust Programming Language

1. Gain essential programming skills with Rust
2. Become a competent system programmer
3. Job opportunities and career advancement
4. Enrich your CV and attract better jobs
5. Increase your earning potential

Rust Programming Language Course Outline

Rust - Introduction
Rust - Environment Setup
Rust - HelloWorld Example
Rust - Data Types
Rust - Variables
Rust - Constant
Rust - String
Rust - Operators
Rust - Decision Making
Rust - Loop
Rust - Functions
Rust - Tuple
Rust - Array
Rust - Ownership
Rust - Borrowing
Rust - Slices
Rust - Structure
Rust - Enums
Rust - Modules
Rust - Collections
Rust - Error Handling
Rust - Generic Types
Rust - Input Output
Rust - File Input/ Output
Rust - Package Manager
Rust - Iterator and Closure
Rust - Smart Pointers
Rust - Concurrency
Rust - Video Lectures
Rust - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

SCALA COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Scala Programming?

Scala which is the short form for **Scalable Programming Language** is a hybrid, multi-purpose, object-oriented, functional programming language. It was developed by Martin Odersky. Scala smoothly added all the features of both object-oriented and functional languages. Scala programs are compiled to run on the Java Virtual Machine. Many already existing software companies, who completely depend on Java for handling their business-critical applications, are turning to Scala to further boost the productivity of their development, applications scalability and overall reliability.

Scala joins basic object-oriented and functional programming in one small, high-level language. Scala's static types help get rid of errors and bugs in large and complex applications, and its Java virtual machine and JS runtimes let you write and build high-performing applications and systems with very simple, easy access to the large ecosystem of libraries.

Scala is a more than a purpose software programming language that provides much support for object-oriented and functional programming methodologies, and it's also built with a powerful static type system. It was designed to be small and concise, most of the decisions made when designing scala were aimed at addressing criticisms of Java.

Scala is a completely object-oriented programming language in the sense that everything in it is an object. Classes in Scala are further extended by subclassing another class together with a very flexible, mixin-based arrangement mechanism as a clean replacement for multiple inheritances as seen in other languages. Scala is also a functional programming language in the sense that every function in Scala is a value and every value is an object, therefore, in Scala, every function is an object.

Scala provides a lightweight and easy to understand the syntax for defining anonymous functions, it offers supports for higher-order functions, it allows functions to be nested and supports currying.

Features of Scala Programming

1. Type Inference: In Scala, you are not required to declare data type and function return type explicitly. Scala is smart enough to find out the type of data. The return type of the function is gotten by the type of the last expression which is present in the function.

2. Singleton object: In Scala, there are no static methods or static variables. Scala makes use of a singleton object, which is basically classed with only one object in the source file. Singleton object is declared in Scala by using the object instead of the class keyword.

3. Immutability: Scala adopts the concepts of immutability. Each variable that is declared is immutable by default. Immutable here means you can't change its value. You can also generate mutable variables that can be modified.

4. Lazy Computation: In Scala, computation is lazy by default. Scala evaluates expressions only when they are required. You can declare a lazy variable by using a lazy keyword. It is used to increase performance.

5. Case classes and Pattern matching: Case classes in Scala are just normal classes that are immutable by default. All the parameters that are listed in the case class are public and immutable by default.

6. Concurrency control: Scala presents you with a standard library that holds the actor model. With Scala, you can write various concurrency codes by using an actor.

7. String Interpolation: Since the release of Scala 2.10.0, Scala presents a new feature to create strings from your input data. It is referred to as string interpolation. String interpolation allows users to include variable references directly in your processed string literals. Scala provides three methods for string interpolation which are: s, f and raw.

Benefits of Scala Programming

Below are some of the benefits of Scala:

1. Scala allows you to use either the Object-Oriented paradigm or Functional Coding patterns.
2. Scala comes with Compatibility and Interoperability for Java programs.
3. Scala has a lot of advanced features.
4. Scala has a pretty good IDE support.
5. Sacla programs are highly scalable.
6. Scala ensures that you write high-quality code.
7. Learning scala will help you command a very high paying job because scala skilled people are being sorted after by the giant tech companies like Netflix, Airbnb, Morgan Stanely.
8. Scala works well for large-scale applications and data engineering infrastructure.

Why Study Scala Programming?

1. Scala is a pure object-oriented language.
2. Scala is also a functional language.
3. Scala has it's type system to be static.
4. Scala runs on the Java Virtual Machine.
5. Scala can run and execute java code
6. Scala can perform Synchronous and concurrent code processing.
7. Career opportunities and advancement

Scala Web Frameworks

Scala is mostly being used on every platform and most importantly it is used in enterprise web applications. Some of the web frameworks that allow it to be used on the web include:

1. The Lift Framework.
2. The Play Framework.
3. The Bowler Framework.

Concepts in Scala Programming

1. Objects: Objects possess states and functions. An object is the basic unit of a class.

2. Class: A class can be said to be like the main template that shows the functions or states that are related to just the class.

3. Methods. A method can be said to be a behavior or a function. A class can also contain methods and not just objects. The logic is done in methods, manipulation of data and all actions are executed in methods.

4. Fields: Every single object has its special set of variables which are referred to as fields. An object's behavior is created by the values that were assigned to these fields.

5. Closure: A closure is basically a method or function whose return value solely depends on the value of variables written outside this particular function.

6. Traits: A trait is a hidden function and field, which can be reused by combining them with classes. Traits are often used to state object types by specifying the signature of the supported methods.

Basic Syntax

The following are the basic syntaxes and coding conventions in Scala programming.

1. Case Sensitivity: Scala is a case-sensitive type language, which means Food and food mean different things in scala.

2. Class Naming: In the class naming convention for scala, the first letter should always be in Upper Case. If several words are used to form a single class name then the words should be in camel casing.

3. Method Names: All methods or function names are to be started with a Lower Case letter.

4. Program File Name: The name of the program file should exactly be the same as the object name.

5. def main(args: Array[String]): Scala program processing begins from the main() function which is a very important and unignorable part of every Scala Program.

Scala Programming Course Outline

Scala Programming - Introduction
Scala Programming - Overview
Scala Programming - Environment Setup
Scala Programming - Basic Syntax
Scala Programming - Data Types
Scala Programming - Variables
Scala Programming - Classes & Objects
Scala Programming - Access Modifiers
Scala Programming - Operators
Scala Programming - IF ELSE
Scala Programming - Loop Statements
Scala Programming - Functions
Scala Programming - Closures
Scala Programming - Strings
Scala Programming - Arrays
Scala Programming - Collections
Scala Programming - Traits
Scala Programming - Pattern Matching
Scala Programming - Regular Expressions
Scala Programming - Exception Handling
Scala Programming - Extractors
Scala Programming - Files I/O
Scala Programming - Video Lectures
Scala Programming - Exams And Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

SWIFT PROGRAMMING COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Swift Programming?

Swift or Swift Programming is a multi-paradigm, all-purpose, compiled computer programming language that is developed by Apple Inc. for iPadOS, iOS, macOS, tvOS, watchOS, Linux, and z/OS.

Swift is designed and developed to work with Apple's Cocoa and Cocoa Touch frameworks and the large body of pre-existing Objective-C code that is written for Apple software products. Swift is built with the open-source LLVM compiler framework and it has been included in Xcode since its 6th version, it was released in 2014. On Apple platforms, it makes use of the Objective-C runtime library which allows Objective-C, C, C++, and Swift code to run inside one program

Swift is the product of the latest research on software programming languages, coupled together with decades of experience of building Apple platforms. Named parameters are declared in a clean syntax that makes developing APIs in Swift even easier to make use of and maintain. Even better, with swift, there is no need to type semi-colons. Inferred types make code to be cleaner and less prone to mistakes, while modules remove the headers and offer namespaces.

Apple proposed Swift to offer support for many-core concepts that are associated with the Objective-C programming language, notably dynamic dispatch, extensible programming, widespread late binding, and similar features, but in a "safer" way, therefore making it very easy to catch bugs in the software; Swift has various features that address some basic programming errors such as null pointer dereferencing and it provides syntactic sugar to help developers avoid the monument of doom. Swift promotes the notion of extensibility of protocols, an extensibility system that can be applied to structs, types, and classes, which Apple advertises as a real change in regular programming paradigms they term "protocol-oriented programming" (which is very similar to traits).

Swift 4 comes with a playground feature where software developers can write their code and run it to see the results instantly. The first public release of Swift was published in 2010. It took Chris Lattner almost 14 years to come up with the first public official version, and later, it was supported and sponsored by many other contributors. Swift designers took their ideas from various other popular programming languages such as Rust, Objective-C, Haskell, Ruby, C#, Python, and CLU.

Features of Swift Programming

There are many features of Swift and some of them are:

1. Swift has closures that are unified with function pointers.
2. Swift has tuples and they return multiple values.
3. Swift comes with Generics.
4. Swift has a Fast and concise iteration over a range, list, or collection.
5. Swift has structs that have various support methods, extensions, and protocols
6. Swift has functional programming patterns, e.g, map, and filter.
7. Swift comes with a powerful error handling feature built-in.
8. Swift has an advanced control flow with, defer, guard, and repeat keywords.
9. Swift makes use of safe programming patterns.
10. Swift provides modern programming features.
11. Swift provides Objective-C like syntax.
12. Swift is a fantastic way to write iOS and OS X apps.
13. Swift provides seamless access to existing Cocoa frameworks.

Benefits of Swift Programming

There are many benefits of Swift and some of them are:

1. Swift has an awesome Open-Source Community.
2. Swift codes are easy and readable.
3. The maintenance of Swift codes is easy.
4. Swift codes allow for speedy App Development.
5. Swift programs are compatible with Objective-C.
6. Programs that are written with Swift are less prone to errors.
7. Swift programs are supported by multiple devices.

Swift Programming Course Outline

Swift Programming - Introduction
Swift Programming - Overview
Swift Programming - Environment
Swift Programming - Basic Syntax
Swift Programming - Data Types
Swift Programming - Variables
Swift Programming - Optionals
Swift Programming - Tuples
Swift Programming - Constants
Swift Programming - Literals
Swift Programming - Operators
Swift Programming - Decision Making
Swift Programming - Loops
Swift Programming - Strings
Swift Programming - Characters
Swift Programming - Arrays
Swift Programming - Sets
Swift Programming - Dictionaries
Swift Programming - Functions
Swift Programming - Closures
Swift Programming - Enumerations
Swift Programming - Structures
Swift Programming - Classes
Swift Programming - Properties
Swift Programming - Methods
Swift Programming - Subscripts
Swift Programming - Inheritance
Swift Programming - Initialization
Swift Programming - Deinitialization
Swift Programming - ARC Overview
Swift Programming - Optional Chaining
Swift Programming - Type Casting
Swift Programming - Extensions
Swift Programming - Protocols
Swift Programming - Generics
Swift Programming - Access Control
Swift Programming - Video Lectures
Swift Programming - Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

VISUAL BASIC.NET COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What Is Visual Basic.NET?

Visual Basic.NET Framework which is mostly shortened to as VB.Net is a simple yet modern, object-oriented computer programming language that is developed by Microsoft to merge the power of .NET Framework and the CLR (common language runtime) together ith the productivity benefits that are the trademarks of Visual Basic.

VB.NET is an object-oriented software programming language that is executed on top of the .NET Framework. Although it is an evolution of standard Visual Basic language, it is non-backward-compatible with Visual Basic 6 and any other code that is written in the old version can not compile under VB.NET.

Just like all other .NET languages, Visual Basic.NET has complete support for object-oriented programming concepts. Everything in VB.NET is treated as an object, including all of its primitive data types e.g Short, Long, Integer, String, Boolean, etc. and also,user-defined types, events, and even assemblies. All objects in VB.NET inherits from the base class Object.

VB.NET is executed by Microsoft's .NET framework. This Implementation then gives it full access to all the libraries that are present in the .Net Framework. It's also possible to run VB.NET programs on Mono software, which is the open-source alternative to the .NET framework, not only under Windows OS but also on Linux or Mac OSX.

The .Net Framework applications are generally multi-platform applications. The .NET framework has been designed in such a way that it can be used from any of the following languages: Visual Basic, C++ C#, Jscript, COBOL, etc. All these languages can directly access the framework as well as communicate with each other.

The .Net Framework has an enormous library of codes that are used by other client languages such as VB.Net. These languages make use of the object-oriented methodology.

Features Of Visual Basic.NET

1. VB .Net incorporates various features like the in-place menu editor, automatic control anchoring and docking, etc, all of which help web developers to develop web application software easily and faster.

2. The Visual Basic.NET programming language implements automatic formatting of code, a highly intelligent and improved object browser, an XML creator and lots more features.

3. By using the VB.NET Programming Language, web developers can easily develop web applications that have features like event log, file system, performance counters, etc.

4. Since the VB.Net programming language is an object-oriented programming language, web developers can create enterprise-class software using its full object-oriented constructs.

5. Visual Basic.NET programming language provides support for various language features like the implementation of inheritance, polymorphism, encapsulation, etc

6. Visual Basic .NET presents a visual designer for forms to help web developers develop web forms easily. Besides this, web developers can also use its "drag and drop" feature to drag and reposition elements according to their business requirements and needs.

Benefits Of VB.NET

1. Object-Oriented: All data in VB.Net is primarily considered as an object. From its primitive data types to user-defined data types each and everything in VB.NET is object-based. It easily presents you with all the advantages of any OOP's based language like Java, Python, C++, etc.

2. Modern and General Purpose: VB.Net is a general-purpose web programming language that has all the features of any modern web development technology incorporated in it.

3. Easy to Learn: Due to it's very easy to understand syntax and easier learning curve, it is easier to learn the VB.NET programming language.

4. Structured Language: VB.Net is a highly structured programming language.

5. Platform Independent: VB.NET is a platform-independent computer programming language therefore, it could be compiled on lots of computers and platforms and can be run on different OS.

6. Large Community Support: Since VB.NET is par is a part of .Net Framework it generally inherits from its large community support.

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

VISUAL STUDIO.NET COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is VisualStudio.NET?

Visual Studio.NET is an Integrated Development Environment (IDE), built and managed by Microsoft that can be used for developing all kinds of consoles, Windows Forms, graphical user interfaces (GUIs), Mobile Applications, Web services, and Web applications.

Visual Studio is used to develop native code and managed code that is supported by the Microsoft Windows, Windows CE, Windows Mobile, .NET Framework, .NET Compact Framework and Microsoft Silverlight. The Visual Studio .NET's code editor offers various supports for IntelliSense and code refactoring, while the Visual Studio .NET integrated debugger supports both adding breakpoints to the source code and machine-level debugging. Visual Studio.NET comes with other built-in tools, like a form designer, which is a very useful tool when developing GUI applications, it is also a very useful tool for all kinds of use cases such as, Web designer whose tasks is to create dynamic Web pages, a class designer that employed to develop custom libraries, and a schema designer for database support.

To improve the functionality of VisualStudio at many levels, third party plug-ins may be added to offer support for source-control systems, while editors that are domain-specific language and visual designers are a few examples of the new toolsets that may be added through these plug-ins.

The Built-in language tools offer support for different well-known languages (like C, C++, C#, Python, and Ruby#), while language services may be separately installed to support languages, including M, Python, and Ruby, among other supported languages.

There are Visual Studio versions that offer support for specific languages, which are a cost-reduced solution for newcomers to development who might only need to use Visual Basic, or for advanced users who work on fast processing software applications and continually need to use Visual C# or Visual C++, but are not interested in the low-speed Visual Basic.

Features of Visual Studio.Net

1. Code Editor: Like every other IDE, VisualStudio .NET comes with a code editor that offers support for syntax highlighting and code completion and also has IntelliSense features for variables, functions, class methods, loops, and LINQ queries. IntelliSense is completely supported for the languages that are included in the IDE, such as, Cascading Style Sheets, XML, and JavaScript when developing web applications and web sites. The code editor of VisualStudio .NET is used for all the languages that it supports.

2. Debugger: Visual Studio .NET comes with a debugger that works both as a debugger for source-level codes and as a machine-level debugger. It operates with both managed code as well as native code and it can be used for debugging applications that are written in any language which is supported by Visual Studio. In addition, it can also be attached to running processes, to monitor, and debug those processes.

3. Designer: VisualStudio .NET includes a list of other visual designers to help with the development of applications. These tools include Windows Forms Designer, Web designer/development, WPF Designer, Data designer, Class designer, and Mapping designer.

4. Support for Git integration.

5. Terminal Support.

6. File Hierarchy Structure.

7. Web-Support.

8. Extensions and Support.

Benefits of VisualStudio.Net

There are many benefits of VisualStudio and some of them are:
1. VisualStudio is a platform-independent IDE which means that it can work on Windows, Linux, Mac, etc.
2. VisualStudio is a light-weight IDE and it is incredibly fast with all its tools.
3. VisualStudio has a robust Architecture.
4. Intelli-Sense is added for all its supported languages.
5. Freeware: VisualStudio is free software, you would not have to pay for any license of any sort.

Tools in VisualStudio.Net

1. Open Tabs Browser.
2. Properties Editor.
3. Object Browser.
4. Solution Explorer.
5. Team ExplorerData Explorer.
6. Server Explorer.
7. PreEmptive Protection-Dotfuscator Community Edition.
8. Text Generation Framework.
9. ASP.NET Web Site Administration Tool.
10. Visual Studio Tools for Office.

Supported Products

1. Microsoft Visual C++
2. Microsoft Visual C#
3. Microsoft Visual Basic
4. Microsoft Visual Web Developer
5. Azure DevOps
6. Visual FoxPro
7. Visual SourceSafe
8. Microsoft Visual J++/Microsoft Visual J#
9. Visual InterDev

Why Study Visual Studio.Net?

1. It has a robust architecture.

2. VisualStudio is a light-weight IDE and it is incredibly fast with all its tools.

3. VisualStudio is a platform-independent IDE which means that it can work on Windows, Linux, Mac, etc.

4. Career Opportunities And Advancement

Visual Studio.Net Course Outline

Introduction
Understanding Visual Studio 2010 Essentials
Learning the VS 2010 Environment
Building Programs with VS 2010
Enhancing the VS 2010 Experience
Video Lectures
Exams and Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

VISUAL BASIC COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Visual Basic?

Microsoft's Visual Basic is basically an event-driven third-generation software programming language that is owned and managed by Microsoft for its Model COM (Component Object Model) programming model.

Microsoft wanted Visual Basic to be very easy to learn and code with. The name Visual Basic was derived from BASIC and it supports for the rapid application development (RAD) of various graphical user interface (GUI) software applications and direct access to databases by making use of Data Access Objects, ActiveX Data Objects or Remote Data Object and the creation of various ActiveX controls and objects.

Like the BASIC Software Programming Language, Microsoft Visual Basic was designed to have an easy learning curve. Software programmers can create and edit both simple and complex GUI programs.

Programming in Visual Basic is a combination of graphically arranging components or controls on a form window, along with defining and specifying the various attributes and actions for those features, and also writing additional lines of code for better performance and functionality.

Visual Basic can be used to develop system executables (.exe files), DILL files or ActiveX controls, but it is basically used to create Windows software and to access database systems.

Features Of Visual Basic

There are lots of features and characteristics of Visual Basic and some of them are;

1. There are no line numbers in Visual Basic as found in earlier BASIC language, codes in VB are grouped into subroutines or methods.

2. In Visual Basic, code Statements have no terminating character except a line-ending character.

3. Code comments in Visual Basic are declared with a single apostrophe (') character. ' This is a defined comment.

4. Looping statement code blocks start and end with keywords like While-End, Do-Loop, For-Next, etc.

5. Logical and bitwise code operators are unified. This is not the case in some languages that are derived from C such as Perl, in which you will have different logical and bitwise operators. This again is a fundamental feature of BASIC.

6. Visual Basic has a variable array base. Arrays are basically defined and declared by stating the upper and lower bounds in a way that is very closely related to languages like Pascal and Fortran. It is also possible to use the Option Base statement in Visual Basic to set the default lower bound.

7. Visual Basic has a relatively powerful integration with the Windows operating system and also, the Component Object Model. The native data types for strings and arrays are the dedicated COM types, SAFEARRAY and BSTR.

Benefits Of Visual Basic

There are many benefits and advantages of Visual Basic and some of them are:

1. The file structure and syntax of the Basic programming language are very simple to work with, especially as to the system executable code.

2. Visual Basic is not only a programming language but it is also basically an integrated and interactive software development environment (IDE).

3. The Visual Basic IDE has been highly built and optimized to support for the rapid application development (RAD). Visual Basic makes it particularly very easy to build and develop graphical user interfaces and to connect them to various handler functions that are provided by the application.

4. The graphical user interface of the Visual Basic IDE provides you with an intuitively and appealing view for the management of the program structure in the various types of entities such as classes, modules, procedures, forms, etc.

5. Visual Basic provides you with a comprehensive interactive and context-sensitive online help system to assist you with your development activities.

6. There is a huge catalog of readily available COM components and libraries for many different purposes in Visual Basic.

Visual Basic Course Outline

Visual Basic - Introduction
Visual Basic - Build Application
Visual Basic - Controls
Visual Basic - Write Code
Visual Basic - VB Data
Visual Basic - Variables
Visual Basic - If and IIf
Visual Basic - Select Case
Visual Basic - Looping
Visual Basic - Functions Intro
Visual Basic - Math Functions
Visual Basic - String Functions
Visual Basic - Sub Procedure
Visual Basic - Excel VBA Functions
Visual Basic - Arrays
Visual Basic - Create Files
Visual Basic - Create Graphics
Visual Basic - Create DVD Player
Visual Basic - Create Audio Player
Visual Basic - Create a Picture Viewer
Visual Basic - Multimedia Player
Visual Basic - Create Database
Visual Basic - Advanced Database
Visual Basic - ADO Control
Visual Basic - DataGrid Control
Visual Basic - SQL Queries
Visual Basic - More SQL Queries
Visual Basic - E-Library
Visual Basic - Animation 1
Visual Basic - Animation 2
Visual Basic - Animation 3
Visual Basic - Web Browser
Visual Basic - FTP Program
Visual Basic - Errors Handling
Visual Basic - Complie/Distribute
Visual Basic - Creating Menu
Visual Basic - Keyboard Handling
Visual Basic - Printing
Visual Basic - Creating a Report
Visual Basic - Video Lectures
Visual Basic - Exams And Certification

BISMARCK

TECHNOLOGY & LEARNING INSTITUTION LIMITED

Every Life, Every Moment, Every Day. A New Discovery...

XAMARIN COURSE

DURATION: 2 WEEKS
FORMAT: WEB/PDF PLUS VIDEO LECTURERS

CERTIFICATE OF COMPLETION

What is Xamarin?

Xamarin is an open-source software development platform for designing and building modern and highly optimized applications platforms such as iOS, Android, and Windows with .NET. framework, Xamarin is basically an abstraction layer that manages the communication of shared code with the underlying platform code. Xamarin runs in a managed environment that provides benefits such as garbage collection and memory allocation.

Xamarin allows software developers to share a percentage of over 90% of their applications across multiple platforms. This design pattern allows software developers to write all of their business logic in just one language (or simply reuse their existing application code) yet being able to achieve pure native application performance, looks, and feel on each platform.

Xamarin applications can be written on either a Windows, Linux or Mac PC and are able to compile into native application packages, such as an .apk file for Android, or an .ipa file on iOS.

With a C#-shared codebase, software developers can use Xamarin tools to design and develop native Android, iOS, and Windows applications that have native user interfaces and also share code over multiple platforms, including Windows, Linux, and macOS.

Features Of Xamarin

Below are some of the features of Xamarin:

1. Xamarin.iOS: Xamarin.iOS allows us to design and develop applications for IOS while using the same UI controls that are made available for other platform apps, except with the suppleness and beauty of a modern language like C#.

2. Xamarin.Android: Xamarin.Android allows software developers to design and develop Android applications while using the same UI controls, except with the power of the.NET Base Library, flexibility, and two first-class IDEs namely, Xamarin Studio and Visual Studio IDE.

3. Xamarin.Mac: Xamarin.Mac allows software developers to design and develop applications for MAC os in C# and.NET framework while using the same OS X libraries and UI design controls. Because Xamarin.Mac is directly integrated with Xcode, the software developer can use Xcode's Interface Builder to create an app's User Interfaces or optionally develop them directly from their C# code.

4. Xamarin Test Cloud: The Xamarin Test Cloud is used for the testing of Xamarin code on various virtual mobile platforms and devices. The Xamarin Test Cloud is simply a cloud-based software solution that provides software developers with various development tools that support automated UI Acceptance Testing of mobile applications across hundreds of different mobile devices. since its cloud-based, the maintenance and performance efforts are removed from the test consumer.

5. HockeyApp: HockeyApp helps you to collect feedback from your users for the app. Xamarin's possible services will also be incorporated into the Microsoft DevOps and enterprise development tools offerings, which provides developers with a comprehensive solution that traverses every phase of the mobile development cycle. We can determine the mixture of the .NET library functionality and the native Mac OS X APIs (application programming interfaces for the ease of the development of any mobile application.

Benefits Of Xamarin

There are lots of benefits of Xamarin and some of them are:

1. With Xamarin, you spend fewer hours developing software, unlike other platforms.
2. With Xamarin you generate codes that can run on multiple platforms.
3. You can develop a flawless UI with Xamarin.
4. Xamarin is based on C#
5. Xamarin offers supports for many libraries
6. Effective offline applications
7. Xamarin matches native standards
8. Xamarin comes with an Integrated testing environment
9. Xamarin saves the monetary cost of development.

Benefits Of Xamarin

Xamarin - Introduction
Xamarin - Installation
Xamarin - First Application
Xamarin - Application Manifest
Xamarin - Android Resources
Xamarin - Android Activity Lifecycle
Xamarin - Permissions
Xamarin - Building the App GUI
Xamarin - Menus
Xamarin - Layouts
Xamarin - Android Widgets
Xamarin - Android Dialogs
Xamarin - Gallery
Xamarin - Andriod Views
Xamarin - Multiscreen App
Xamarin - Deploying Your App
Xamarin - Video Lectures
Xamarin - Exams and Certificatio

Madras Road, St. Helena Village, Piarco, Trinidad W.I.
Tel: 1 (868) 745-7409
www.bismarcktechnology.com
Email: contact@bismarcktechnology.com