Certificate Course In Programming Fundamental(c & C++)

Eligibility:10th
Course Code:CR-11

Duration:2 Month

Fundamentals of Computer

Unit-1: Introduction to Computer and Problem Solving: Information and Data Hardware: CPU, Primary and Secondary storage, I/O devices, Bus structure, Computer Peripherals - VDU, Keyboard, Mouse, Printer.

Unit-2: Software: System and Application. Different System Software. Programming Languages: Machine Language, Assembly Language, High Level Language, Object Oriented Language.

Unit-3: Problem solving: Algorithm, Flow charts, Decision tables & Pseudo codes. Number systems and Codes: Number representation: Weighted codes, Non-weighted codes, Positional, Binary, Octal, Hexadecimal, Binary Coded Decimal (BCD), Conversion of bases. Complement notations, Binary, Arithmetic, Binary Codes: Gray, Alphanumeric, ASCII, EBCDIC, Single Error-Detecting and Correcting Codes, Hamming Codes.

Unit-4: Basic Computer Organization - IAS Computer, Von Neumann Computer, System Bus. Instruction Cycle, Data Representation, Machine instruction and Assembly Language, CPU Organization, Arithmetic and Logic Unit, Control Unit, CPU Registers, Instruction Registers, Program Counter, Stack Pointer.

Unit-5: Introduction to Networking, Advantages of Networking; Basic Features, LAN, MAN and WAN; simple PC Based Network: Example, block diagram. Mode of operation and characteristic features. Intranet and Internet; Servers and Clients; Ports; Domain Name Server (DNS); WWW, Browsers Connections: Guided and Unguided media - Dial up, ISDN, ADSN; Cable, Modem; E-mail, Voice and Video Conferencing, Management Information System, Characteristics Of MIS, Advantages And Disadvantages Of MIS, Computer Application In Business, Computer Application In Project Management, Work

OPERATING SYSTEMS

Unit-1: Disk Operating System (Dos) :-- Introduction, History & Versions Of Dos, Basic Physical Structure Of Disk, Drive Name, Fat, File & Directory Structure And Naming Rules, Booting Process, Dos, System Files, Dos Commands : Internal – Dir, Md, Cd, Rd, Copy, Del, Ren, Vol, Date, Time, Cls, Path, Type Etc. External- Chkdsk, Xcopy, Print, Diskcopy, Diskcomp, Doskey, Tree, Move, Label, Append, Format, Sort, Fdisk, Backup, Edit, Move, Attrib, Help, Sys Etc. Executable V/S Non Executable File In Dos.

Unit-2: Windows XP :-- Introduction To Window XP And Its Features. Hardware Requirements Of Windows. Window Concepts, Window Structure, Desktop, Taskbar, Start Menu, My Pictures, My Music Working, With Recycle Bin–Restoring A Deleted File, Emptying The Recycle Bin. Managing Files, Folders And Disk Navigating Between Folders, Manipulating Files And Folders, Creating New Folder, Searching Files And Folder My Computer Exploring Hard Disk, Copying And Moving Files And Folder from One Drive To Another Formatting Floppy Drive, Windowa Accessories-Calculator, Notepad, Paint, WordPad, Character Map, Paint Command Prompt, Window Explorer- Exploring Hard Disk Coping And Moving Files And Folder From One Drive To Another, Formatting Floppy Drive And Other Explorer Facilities. Burning Cd Entertainment Player, Dvd Player, Media Player, Sound Player, Volume Control, Movie Marker.

Unit-3: Advanced Features Of Window Xp :-- Managing Hardware & Software–Installation Of Hardware & Software Using Scanner Web Camera, Printers, Sharing Of Printers, Seytem Tools–Backup,Character Map, Clipboard Viewer, Disc Defragment, Drive Space, ScanDisk, System Information, System Monitor, Drive Converter, (Fat 32), Disk Cleanup, Using, Windows Update. Communication – Dial Up Networking, Direct Cable Connection, Hyper Terminal, Phone Dial, Browsing The Web With Internet Explorer, Communication Though, Outlook Express, Multiple Users, Features Of Window, Creating And Deleting Users, Changing User ,Password Etc Accessibility Features Of Windows–Sharing Information, Between Program, Sharing Folder And Drive, Browsing The Entire Network, Mapping Windows Shared Drive, Using Shared Printers –Understanding Ole-Embed/ Link Using Insert Object – Manage Embedded /Linked Object.

Unit-4: LINUX :-- History & Features Of Linux, Linux structure, file System of linux hardwere requirement of linux, various flavource of linux, program & process, process creatior and process identifiers, functions of profile cnd login file in linux, linux kernel, Multi user features of linux, login and logout from linux system linux command –bc, cal, cat, mv, rm, rmdir, tty,wc, who, whois, grep, white, telnet, pipeline concepts using floppy and cd rom in linux vi editor.

Unit-5: Windows Fundamental :-- Windows is an operating system, XP Home Addition, XP Professional Addition, Active desktop, Multi user account, Data protection features, Playing MP3 Sound and XP Various version of Windows :-- Windows desktop fundamental, Desktop, Taskbar, Icons, Recycle bin, Folder.

Principles of Programming Techniques

`

The Principles of Programming research group at NUI Maynooth specialises in the static and dynamic analysis of object-oriented programs and programming languages. We exploit a variety of techniques, such as parsing, bytecode analysis, software metrics, meta-modelling and program verification to model software systems in order to increase comprehensibility and reliability. Our work has applications in reverse engineering, program verification and validated forward engineering from design to code.

Need & Type Of Computer Language

Language Types

Procedural

The programming style you're probably used to, procedural languages execute a sequence of statements that lead to a result. In essence, a procedural language expresses the procedure to be followed to solve a problem. Procedural languages typically use many variables and have heavy use of loops and other elements of "state", which distinguishes them from functional programming languages. Functions in procedural languages may modify variables or have other side effects (e.g., printing out information) other than the value that the function returns.

Functional

Employing a programming style often contrasted with procedural programming, functional programs typically make little use of stored state, often eschewing loops in favor of recursive functions. The primary focus of functional programming is on the return values of functions, and side effects and other means storing state are strongly discouraged. For instance, in a pure functional language, if a function is called, it is expected that the function not modify any global variables or perform any output. It may, however, make recursive calls and change the parameters of those calls. Functional languages are often simpler syntactically and make it easier to work on abstract problems, but they can also be "further from the machine" in that their programming model makes it hard to understand exactly how the code is translated into machine language (which can be problematic for system programming).

Object-oriented

Object-oriented programming views the world as a collection of objects that have internal data and external means of accessing parts of that data. The goal of object-oriented programming is to think about the problem by dividing it into a collection of objects that provide services that can be used to solve a particular problem. One of the main tenets of object oriented programming is encapsulation -- that everything an object will need should be inside the object. Object-oriented programming also emphasizes reusability through inheritance and the ability to extend current implementations without having to change a great deal of code by using polymorphism.

Scripting

Scripting languages are often procedural and may contain elements of object-oriented languages, but they fall into their own category because they are typically not meant to be full-fledged programming languages with support for large system development. For instance, they may not have compile-time type checking or require variable declarations. Typically, scripting languages require little syntax to get started but make it very easy to make a mess.

Logic

Logic programming languages allow programmers to make declarative statements (possibly in first-order logic: "grass implies green" for example) and then allow the computer to reason about the consequences of those statements. In a sense, logic programming is not telling the computer how to do something, but placing constraints on what it should consider doing.

Compilar, Interpreter & Assembler

A compiler is responsible for translating a source code, which is written in a programming language, into a target language. This is most commonly done in order to create an executable program. A compiler is mainly used for programs that translate a source code into an assembly language or machine code, which are both a lower level language than the source code. An interpreter is a program that carries out the instructions that are written in a programming language. It can do this in a number of ways: It can directly perform the source code; it can decipher the source code into an efficient code (also known as intermediate representation) and implement this straight away; or it specifically carries out a recompiled code that has been made and stored by a compiler. Assemblers create an object code by translating assembly instruction mnemonics into opcodes. They also determine symbolic names for memory locations as well as for other entities. A prime characteristic of assemblers is the use of symbolic references, which saves time consuming manual calculations and address updates after a program has been modified. The majority of assemblers also have macro facilities so that they can perform textual substitution, which means that they are able to create short sequences of instructions.

C/C++ Feature & Programming Concepts

Unit-1: C++ Characteristics, Object-Oriented Terminology, Object-Oriented Paradigm, Abstract Data Types ,I/O Services Standard Template Library Standards Compliance, Functions and Variables, Functions: Declaration and Definition Variables: Definition, Declaration, and Scope, Variables: Dynamic Creation and Derived Data, Arrays and Strings in C++, Qualifiers, Classes in C++, Defining Classes in C++,Classes and Encapsulation, Member Functions, Instantiating and Using Classes, Using Constructors, Multiple Constructors, Parameter Constuctor, Copy Constructor and Initialization Lists, Using Destructors to Destroy Instances Friendship, Input and Output in C++ Programs .

Unit-2: Operator Overloading, Operator Overloading, Working with Overloaded Operator Methods ,Initialization and Assignment, Constant and Static Class Members, Dynamic Allocation: new and delete ,Single Inheritance, Multiple Inheritance, Multilevel Inheritance, Heiarchical Inheritance, Hibrid Inheritance, Defining Base and Derived Classes, Polymorphism, Overview of Polymorphism .

Unit -3: Standard Streams, Manipulators, Unformatted Input and Output, File Input and Output, Exceptions ,Exception Hierarchies, Inside an Exception Handler .

Unit-4: Flow charts, Tracing flow charts, Problem solving methods, Need for computer Languages, Sample Programs written in C , C character set, Identifiers and keywords, Data types, Declarations, Expressions, statements and symbolic constants, Arithmetic, unary, logical, bit-wise, assignment and conditional operators, While, do-while, for statements, nested loops, if else, switch, break, Continue, and goto statements, comma operators, Automatic, external, register and static variables.

Unit-5: Defining and accessing, passing arguments, Function prototypes, Recursion, Library functions, Static functions, Defining and processing, Passing arrays to a function, Multi dimensional arrays.

Declarations, Passing pointers to a function, Operations on pointers, Pointer Arithmetic, Pointers and arrays, Arrays of pointers function pointers. Defining and processing, Passing to a function, Unions, type def, array of structure, and pointer to structure. Definitions, concept of record, file operations: Storing, creating, retrieving, updating Sequential, relative, indexed and random access mode, Files with binary mode(Low level), performance of Sequential Files, Direct mapping techniques: Absolute, relative and indexed sequential files (ISAM) concept of index, levels of index, overflow of handling. File operation: creation, copy, delete, update, text file, binary file.

C Programming

Unit-1: Algorithem, Flow charts, Tracing flow charts, Problem solving methods, Need for computer Languages, Sample Programs written in C .

Unit-2: C character set, Identifiers and keywords, Data types, Declarations, Expressions, statements and symbolic constants, Arithmetic, unary, logical, bit-wise, assignment and conditional operators, While, do-while, for statements, nested loops, if else, switch, break, Continue, and goto statements, comma operators, Automatic, external, register and static variables.

Unit-3: Function, Defining and accessing, passing arguments, Function prototypes, Recursion, Library functions, Static functions, Defining and processing.

Unit-4: Passing arrays to a function, Array, Multi dimensional arrays , Passing pointers to a function, Operations on pointers, Pointer Arithmetic, Pointers and arrays, Arrays of pointers function pointers. Defining and processing, Passing to a function, Unions, type def, array of structure, and pointer to structure

Unit-5: Definitions, concept of record, file operations: Storing, creating, retrieving, updating Sequential, relative, indexed and random access mode, Files with binary mode(Low level), performance of Sequential Files, Direct mapping techniques: Absolute, relative and indexed sequential files (ISAM) concept of index, levels of index, overflow of handling. File operation: creation, copy, delete, update, text file, binary file.

C++ Programming

Unit-1: C++ Characteristics, Object-Oriented Terminology, Object-Oriented Paradigm, Abstract Data Types ,I/O Services Standard Template Library Standards Compliance, Functions and Variables, Functions: Declaration and Definition Variables: Definition, Declaration, and Scope, Variables: Dynamic Creation and Derived Data, Arrays and Strings in C++, Qualifiers, Classes in C++, Defining Classes in C++, Classes and Encapsulation, Member Functions, Instantiating and Using Classes

Unit-2: Using Constructors, Multiple Constructors, Parameter Constuctor, Copy Constructor and Initialization Lists, Using Destructors to Destroy Instances Friendship, Input and Output in C++ Programs .

Unit-3: Operator Overloading, Operator Overloading, Working with Overloaded Operator Methods ,Initialization and Assignment, Constant and Static Class Members, Dynamic Allocation: new and delete ,Single Inheritance, Multiple Inheritance, Multilevel Inheritance, Heiarchical Inheritance, Hibrid Inheritance, Defining Base and Derived Classes.

Unit -3: Polymorphism, Overview of Polymorphism-runtime and compile time polymorphism.

Unit-4: Function-inline and friend etc.

Unit-5: Standard Streams, Manipulators, Unformatted Input and Output, File Input and Output, Exceptions ,Exception Hierarchies, Inside an Exception Handler.

Project