# Department of Information Technology

Gist of Lectures as per syllabus

 Session 2017-2018

**Honors Paper CC1 - Introduction to Information Technology**

**RATNA PANDEY**

**Asst. Professor (Temp), Department of IT**

**Topic 1: Basic of computer**

Computer is a multipurpose electronic device that can retrieve process & store data. They are used as tools in every part of society. The characteristics of computers that have made them so powerful & universally useful are speed, accuracy, diligence, versatility storage capacity. Computers have gone through different phases of development ie. Different generations.

**Topic 2:Computer Architecture**

It describes a set of rules & methods that describe functionality, organisation & implementation of computer system. Basically a computer system comprises of different input output units & CPU. Various units of the system like I/O units, memory & CPU intercommunicates with each other.

**Topic 3: Input/Output peripherals**

Different units like keyboard, mouse, speech recognition units, scanners & plotters are used to input data in computers. Different devises are used to display output. Output units can be either soft copy devices or hardcopy.

**Topic 4: Computer Software**

Software is a part of computer system that consists of data or computer instructions. Software & Hardware parts of computer are closely related to each other & can't cope up without each other. Before doing anything in computer we develop program which is a set of instructions to do a specific job. These set of steps is known as algorithm. A flowchart is a type of diagram that represents an algorithm. Different programming languages have developed to write programs in computers.

**Topic 5: Open Source Terminologies**

Open Source Software is computer software with its source code made available with a license in which the copyright holder provides the rights to study, change & distribute. Freeware is a computer software distributed under terms that allow users to run the software for any purpose. Shareware is software that is distributed free on a trial basis with understanding that user may need to pay for it later.

**Topic 6: Advanced Trends in IT**

Wireless is the transfer of information between two or more points that are not connected by an electrical conductor. Wi-Fi is a technology for wireless local area networking with devices based on the IEEE 802.11 standards. Bluetooth is a wireless technology standard for exchanging data over short distances from fixed and mobile devices & building personal area network. Social networking service is an online platform that people use to build social relations with each other.

**Department of Information Technology**

Gist of lectures as per syllabus

**Session 2017 – 2018**

**Honors Paper CC4 – Data Structure using C++**

**Topic 1: Basics of C++ program**

C++ is object oriented programming language developed in order to overcome the problems aroused in procedural language. It is an increment of C language. It includes Keywords, Identifiers, Constants, data types etc. C++ adds a number of OOPS features such as objects, inheritance, function overloading & operator overloading to C which enabled it to build programs with clarity & ease of maintenance.

**Topic 2: Classes**

A class is an abstract user defined data type which implements the concept of objects in programming. It is a way to bind data & functions together. Constructors are special member functions having its name same as the class name whose task is to initialise the objects of its class. Destructors are also special member function similar to constructors except its name is preceded by a tilde (~) operator. It destroys the objects created by constructors.

**Topic 3: linked list**

A linked list is a linear collection of data elements, in which linear order is not given by their physical placement in memory. Instead, each element [points](https://en.wikipedia.org/wiki/Pointer_%28computer_programming%29) to the next. It is a [data structure](https://en.wikipedia.org/wiki/Data_structure) consisting of a group of [nodes](https://en.wikipedia.org/wiki/Node_%28computer_science%29) which together represent a [sequence](https://en.wikipedia.org/wiki/Sequence). They can be used to implement several other common [abstract data types](https://en.wikipedia.org/wiki/Abstract_data_type), including [lists](https://en.wikipedia.org/wiki/List_%28abstract_data_type%29), [stacks](https://en.wikipedia.org/wiki/Stack_%28abstract_data_type%29), [queues](https://en.wikipedia.org/wiki/Queue_%28abstract_data_type%29). These can be either singly linked, doubly linked or circular linked list. Different functions are defined to insert, delete & traverse a linked list.

**Topic 4: Stacks**

Stack is collection of elements that follows the LIFO order. **LIFO stands for Last In First Out**, which means element which is inserted most recently will be removed first. A stack has a restriction that insertion and deletion of element can only be done from only one end of stack and we call that position as top. The element at top position is called top element. Insertion of element is called PUSHand deletion is called POP.

**Topic 5: Queues**

Queue is a data structure that follows the FIFO principle. FIFO means First In First Out i.e the element added first in the queue will be the one to be removed first. Elements are always added to the back and removed from the front. It can be implemented either using array or linked list.

**Topic 6: Trees and Graph**

A tree is a widely used [abstract](https://en.wikipedia.org/wiki/Abstract_data_type) [data structure](https://en.wikipedia.org/wiki/Data_structure) that simulates a hierarchical [tree structure](https://en.wikipedia.org/wiki/Tree_structure), with a root value and [sub trees](https://en.wikipedia.org/wiki/Subtrees) of children with a parent node, represented as a set of linked [nodes](https://en.wikipedia.org/wiki/Vertex_%28graph_theory%29). A tree data structure can be defined [recursively](https://en.wikipedia.org/wiki/Recursion)  as a collection of [nodes](https://en.wikipedia.org/wiki/Node_%28computer_science%29) starting at a root node. A tree is a data structure made up of nodes or vertices and edges without having any cycle. A graph data structure consists of a finite [set](https://en.wikipedia.org/wiki/Set_%28computer_science%29) of vertices or nodes or points, together with a set of unordered pairs of these vertices for an undirected graph or a set of ordered pairs for a directed graph. These pairs are known as edges, arcs, or lines for an undirected graph and as arrows, directed edges, directed arcs, or directed lines for a directed graph.

**Topic 7: Searching and Sorting**

Searching is a process of locating a particular element present in a given set of elements. The element may be a record, a table, or a file. In Linear Search the list is searched sequentially and the position is returned if the key element to be searched is available in the list, otherwise -1 is returned. The search in Linear Search starts at the beginning of an array and move to the end, testing for a match at each item. Sorting refers to ordering data in an increasing or decreasing fashion according to some linear relationship among the data items. Sorting can be of different types:

Quick Sort- In this sort an element called pivot is identified and that element is fixed in its place by moving all the elements less than that to its left and all the elements greater than that to its right.

**Insertion sort- is a simple sorting algorithm that builds the final sorted array (or list) one item at a time.** **The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. Bubble sort**

 **is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order.**

# Department of Information Technology

Gist of Lectures as per syllabus

 Session 2017-2018

Honors Paper **VII-Programming using C++**

 **Group A:**

# Topic 1: Principles of Object Oriented Programming

 Object oriented programming is a programming language model organized around objects rather than actions and data rather than logic. It was developed to remove the problems encountered in procedural programming like lesser control on data, inability to develop real world programming

**Topic 2: Concept of C++:**

C++ adds a number of OOPS features such as objects, inheritance, function overloading & operator overloading to C which enabled it to build programs with clarity & ease of maintenance. Tokens in C++ include Keywords, Identifiers, Constants, Strings, and Operators. It includes different Data Types like Built in data types; User defined type, Derived data types.

 **Topic 3: Functions in C++**

Function prototyping describes the function interface to the compiler by giving details such as number & type of arguments & the type of return type. It is compulsory in C++ .Call by value &Call by reference: In call by value the called function creates a new set of variables & copies the values of arguments into them ie. Changes are not made in actual variables. In call by reference the address of the actual variables are passed & so changes are in the actual variables.

 **Topic 4: Classes and Objects**

A class is a way to bind data & functions together. It is an abstract data type. It is specified in two parts ie. Class declaration & Class function definitions. Variables of class called objects are created and using objects class members are accessed .This unit deals with static variables & functions , friend functions.

 **Topic 5: Constructors and Destructors**

Constructors are special member functions having its name same as the class name whose task is to initialise the objects of its class. There are different types of constructors Default Constructor, Parameterised Constructor, Copy Constructor. This unit also explain constructor overloading, Dynamic Constructors & Destructors

**Topic 6: Inheritance**

The mechanism of deriving a new class from old one is called inheritance. The old class is called the base class & the new class is called derived class. There are different types of Inheritance: Single Inheritance, Multiple Inheritance, Hierarchical Inheritance, and Multilevel Inheritance & Hybrid Inheritance. This concept implements the concept of reusability in C++.

**Group B:**

**Topic 1: Pointers**

Pointers to objects: It is possible to take the address of a member of a class & assign

It to a pointer. A class member pointer can be declared using the operator::\* with class

Name.We can also create the objects using pointers & new operator. This unit deals with the concept of this Pointer.

**Topic 2: Polymorphism**

Polymorphism means one name having multiple forms. This unit deals with different types of Polymorphism: Compile time polymorphism & Run time polymorphism in which functions will be selected while the program is running. It is implemented using a mechanism known as virtual function.

Top**ic 3:Files & streams**

A stream is a sequence of bytes & serves as a source or destination for an I/O data. The source stream that provides data to the program is called input stream & the destination stream that receives output from the program is called output stream. A file is a collection of related data stored in a particular area on the disk. Read & write operations can be performed on these files using programs. A file can be opened in two ways: i) Using the constructor function of the class. ii)Using the member function open() of the class.

**Topic 4: Templates**

Templates enable us to define generic classes & functions & thus provide support for generic programming. Template classes & functions eliminate code duplication for different types & thus make the program development easier & manageable.

**Topic 5: Exception Handling**

Exceptions are problems that a program may encounter at run time. Exceptions are of two types: synchronous & asynchronous. C++ provides mechanisms for handling synchronous exceptions. An exception is caused by a faulty statement in a **try** block. The statement discovers the error and throws it, which is caught by a **catch** statement.