CHAPTER – 1

OBJECT ORIENTED CONCEPTS USING C++

1. Define Object or what is the Significance of an Object?
   - Object is a kind of self-sufficient subprogram with a specific functional area.
   - Object is a group of related data and functions that serves those functions.

2. Define Encapsulation
   - The mechanism by which data and functions are bounded together within an object definition is called Encapsulation.

3. Define Polymorphism
   - The ability of an object to respond differently to different messages is called Polymorphism.
   - Example:
     - draw (side) – is defined to draw square.
     - draw (length, breadth) – is defined to draw rectangle.
   - The Function draw( ) accepts different inputs and performs different functions.

4. Define Inheritance
   - The process of acquiring the Base class properties to the derived class is called Inheritance.
   - Animals are called the base class, and Mammals and Birds are called derived classes.
   - The derived classes are power packed, as they include the functionality of the base class along with their own unique features.

5. What are the Advantages of Object Oriented Programming?
   - Class data type allows programs to organize as objects that contain both data and functions.
   - Polymorphism reduces software complexity, as multiple definitions are permitted to an operator or function.

CHAPTER – 2

OVERVIEW OF C++

1. Define Tokens. (Or) What is Token? Classify it?
   - A token is the smallest individual unit in a program. Tokens are classified as below:

   **TOKENS**
   - Keywords
   - Identifiers
   - Constants
   - Operators
   - Punctuators
2. **Define Keywords. (Or) What is meant by Keyword?**
   - **Keywords** have special meaning to the language compiler.
   - These are **reserved** words for special purpose.
   - These words cannot be used as **normal identifiers**.
   
   **Eg:** Auto, Local

3. **What is Identifier? (Or) Define Identifier**
   - **Identifiers** are also called as variables.
   - Variables are **memory** boxes that hold **data** or **values** or **constants**.
   - A variable name can contain Alphabets, Numbers, and only one special character called Underscore.
   - A variable name must begin with Alphabet, or Underscore. But it must not begin with Number.
   
   **Eg:**
   
<table>
<thead>
<tr>
<th>VALID IDENTIFIERS</th>
<th>INVALID IDENTIFIERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>int</td>
</tr>
<tr>
<td>_c1</td>
<td>1c</td>
</tr>
<tr>
<td>Abc</td>
<td>abc@</td>
</tr>
</tbody>
</table>

4. **What is Escape Sequence?**
   - **Escape sequences** are represented using characters prefixed with a backslash.
   - **Escape sequences** are also called as **Non-graphic character constants**.
   
   **Eg:** \a, \b

5. **What is String Literal (or) Define String Literal**
   - **String Literal** is a set of characters surrounded by **double quotes**.
   - **String literals** are treated as **Array of characters**.
   - Each string literal is by default added with a special character ‘\0’ which marks the **end of a string**.
   
   **Eg:** “Computer”

6. **What are the types of Operators based on Operands?**
   - Unary operators require **one** operand
   - Binary operator requires **two** operands
7. Define Conditional Operator (or) what is Ternary Operator?
✓ (?:) is called the ConditionalOperator
✓ It is also known as TernaryOperator

Syntax:
E1? E2: E3 where E1, E2, E3 are all operands.

Example: Output:
int a=4, b=6, c;
10
c=(a<b)?a+b:a-b;
cout<<c;

8. What is Punctuator?
✓ Punctuators are characters to do a specific function.

<table>
<thead>
<tr>
<th>PUNCTUATORS</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>;</td>
<td>Terminates a C++ statement</td>
</tr>
<tr>
<td>//</td>
<td>Treat the Statement as a Single Line Comment</td>
</tr>
</tbody>
</table>

9. What are the categories in the Data types?
✓ They can be broadly classified into the following three categories.
  • User defined type – Structure, Union, Class, & Enumeration
  • Built-in type – Integral, Void and Float
  • Derived type – Array, Function, Pointer and Reference

10. What are the Storage Specifiers in C++?
✓ Storage Specifier is a Qualifier which can be added before the Variable Declaration.
✓ The four storage specifiers are
  • Auto,
  • Static,
  • Extern and
  • Register.

11. What is the purpose of Void data type?
✓ Void data type indicates that it holds nothing.
✓ To indicate the function does not return a value
12. Define Void data type
   - Void data type indicates the compiler that the function does not return a value,
   - Void data type indicates that it holds nothing.
   - To declare a generic pointer

13. Define Pointer variable with Example. (Or) what is meant by Pointer Variable?
   - It is a variable that holds a memoryaddress of another variable.
   - It is not used to store the Data or Value
   - It is used to access the value of another variable by indirectly accessing the memory.
   - It uses two Operators (& - Address of Operator, * - Value at Operator)
     Eg: int *a, b; b = 10; a = &b; cout << *a;

14. What is the usage of Asterix (*) Operator?
   - Used to declare a Pointer variable.
   - Used to display the contents stored at a location (Value at the Operator)
   - It is a Unary operator

15. Define Variable
   - Variables are user defined named entities of memorylocations that can store data.
   - Variables are also known as Identifiers.
     Eg:
     int num;
     num = 5;

16. Give the Syntax of Declaring a Variable with an Example?
    Syntax:
    Data type Space Variable1, Variable2, Variable3.............. Variablen;
    Eg:
    int a;
float b;
char c;

17. What is the Impact of Modifiers?

- **Unsigned** modifier modifies the range of the integer values as the sign bit is also used to store data.
- **Long** increases the bytes for a particular data type, and then increase the range of values.

18. What is meant by Implicit Conversion? (Any Four Points)

- If one operand is of type long double, then the other value is also converted to long double.
- If one operand is of type double, then the other value is also converted to double.
- If one of the operands is a float, the other is converted to a float.
- If one of the operands is an unsigned long int, the other is converted to unsigned long int.
- If one of the operands is a long int, then the other is converted to long int.
- If one of the operands is an unsigned int, then the other is converted to an unsigned int.

19. Define `sizeof` Operator

- It returns the memory size in terms of bytes, of the given expression or data type.

Eg:

```c
int a;
a=4;
cout<<sizeof(a);
```

20. Define Operator and Operand

- Operand is entity which is going to act on the operator.
- Operator is used to specify an operation to be performed.

Eg:

```c
c=a+b; // Here a and b are operands and then + is an operator.
```

21. Define Association (or) what is meant by Association?

- The Operators and Operands are grouped in a specific logical way for evaluation is called as **Association**.

<table>
<thead>
<tr>
<th>OPERATORS</th>
<th>ASSOCIATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>*, /, %</td>
<td>Left to Right</td>
</tr>
<tr>
<td>+, -</td>
<td></td>
</tr>
</tbody>
</table>
22. What is Logical Operator? Give an Example?

- It is used to compare two relational expressions

<table>
<thead>
<tr>
<th>OPERATORS</th>
<th>OPERATOR NAME</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;&amp;</td>
<td>Logical AND</td>
<td>(a&lt;b) &amp;&amp; (a&lt;c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!</td>
<td>Logical NOT</td>
<td>a!b</td>
</tr>
</tbody>
</table>

23. How to Declare a Pointer Variable?

**Syntax:**
```
Datatype Space *VariableName;
```

**Eg:**
```
int *a; // a is a pointer variable of the type int.
float *b; // b is a pointer variable of the type float.
```

24. Write a note on Assignment Operators?

- `=` (Equal to) is called the Assignment Operator.
- The Assignment operator is used to take the value from the Right Hand Side of the Equal to (=) to the Left Hand Side Variable.
- The Right Hand Side may be a value or constant or an expression.

**Eg:**
```
int a=4; // The value 4 is assigned to the Variable a.
```

25. What is Relational Operator? Give an Example?

- It is used to compare values. It will give the result either TRUE or FALSE i.e., 1 or 0.

**Eg:** if a=4, b=6

<table>
<thead>
<tr>
<th>OPERATORS</th>
<th>OPERATOR NAME</th>
<th>EXAMPLE</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>Less than</td>
<td>a&lt;b</td>
<td>True (1)</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or Equal to</td>
<td>a&lt;=b</td>
<td>True (1)</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
<td>a&gt;b</td>
<td>False (0)</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or Equal to</td>
<td>a&gt;=b</td>
<td>False (0)</td>
</tr>
<tr>
<td>==</td>
<td>is Equal to</td>
<td>a==b</td>
<td>False (0)</td>
</tr>
<tr>
<td>!=</td>
<td>Not Equal to</td>
<td>a!=b</td>
<td>True (1)</td>
</tr>
</tbody>
</table>

26. Define Typedef

- Typedef is a User-defined Data typeIdentifier.
Users can define a variable that would represent an Existing data type.

Syntax:
```c
typedef datatype userdefined datatype identifier;
```

Eg:
```c
typedef int a; // a is a user-defined identifier
a a1, a2;
```

27. Define Enum

- Enumerated is also a User-defined data type.
- Enumerated data type helps users in creating a list of identifiers also called as Symbolic Numeric Constants.

Syntax:
```c
enum datatype identifier(value1,value2,.........valuenum);
```

Eg:
```c
enum workdays(Monday,Tuesday,Wednesday,Thursday,Friday);
```

28. Define Auto Variable

- Defines local variable known to the block in which they are defined.
- By default the local variables are auto.

Eg:
```c
void main( )
{
    auto int a;
}
```

29. Define Type cast

- Type cast refers to the process of changing the data type of the value stored in a variable.
- The statement `(float) 7`, converts the numeric constant 7 to float type.
- Type cast is achieved by prefixing the variable or value with the required data type.

CHAPTER – 3

BASIC STATEMENTS

1. Define Header File

- A header file comprises of all standard declarations and definitions for predefined functions.
- A preprocessor directive starts with #, which instructs the compiler to do the required job.
2. Define get from operator (or) what is meant by Extraction Operator?

- The >>is the Extraction or Get from operator.
- It takes the value from the stream object to its left and places it in the variable to its right.

Eg:

```cpp
int a;
cin>>a;
```

3. Define put to operator (or) what is meant by Insertion Operator?

- The operator <<is called the Insertion Operator or Put to operator.
- It directs the contents of the variable to its right to the object to its left.

Eg:

```cpp
int a=4;
cout<<a;
```

4. What are the sections of C++ program?

- Include files
- Declaration of variables, data type, and user defined functions.
- Main() function

5. What is Declaration Statement?

- Declaration of a variable introduces a variable’s name and its associated data type.

Eg:

```cpp
int a; //Here the variable a is introduced and it is data type is specified as Integer.
```

6. What is Control Structure?

- Program statements that cause a jump of control from one part of a program to another are called Control Structures.
- Control Structures are classified into two. They are,
  - Decision – Making Statements,
  - Looping Statements.

7. Define Simple If Statement

- If Statement is the simplest of all the decision statements.

Syntax:

```cpp
if(condition/expression)
```
### 8. What is If – Else Statement?

- If – Else Statement chooses between the two alternatives, executes the chosen block based on the condition.

**Eg:**

```c++
int a = 4, b = 2;
if (a > b)
{
  cout << "The Value of a = " << a;
}
else
{
  cout << b;
}
```

**Output:**

The Value of a = 4

### 9. Define Assignment Statement

- `=` (Equal to) is called the Assignment Operator.
- In Assignment Statement, the Assignment operator is used to take the value from the Right Hand Side of the Equal to (=) to the Left Hand Side Variable.
- The Right Hand Side may be a **value** or **constant** or an **expression**.

**Eg:**

```c++
int a = 4; // The Value 4 is assigned to the Variable a.
int b = 6;
int c = b; // The Variable b is assigned to the Variable c.
int d = a + b; // The Expression is assigned to the Variable d.
```

### 10. Define Loop

- **Loops** execute a set of instructions repeatedly for a certain number of times.
There are two types of Loops

- Entry Check Loop: Eg: While
- Exit Check Loop: Eg: Do-While

11. What are the Kinds of Loop?
- There are three kinds of Loop. They are,
  - While Loop – Entry Check Loop.
  - Do-while Loop – Exit Check Loop.
  - For Loop – Entry Controlled Loop.

12. What is Continue Statement?
- The Continue statement forces the next iteration of the loop to take place, skipping any code following the continue statement in the loop body.

13. What is Break Statement?
- Break statement would exit the current loop only.
- Break statement accomplishes jump from the current loop.

14. What are the Rules for the Nested Loop?
- An outerloop and innerloop cannot have the samecontrolvariable, as it will lead to logical errors.
- The innerloop must be completely nestedinside the body of the outerloop.

CHAPTER – 4
FUNCTIONS

1. Define Function
   - It is the Building blocks of C++ programs. It is also the executable segments in a program. The starting point for the execution of a program is main().

2. What are the Advantages of the Function?
   - Reduce the size of the program
   - Induce the reusability of code.

3. Define Parameters
   - **Parameters** are the channels where the data flows from the **callstatement** to the **functiondefinition** and vice versa.
   - Parameters are of two types. They are,
     - **Formal Parameters**,  
     - **Actual Parameters**.

4. What are the rules for the Actual Parameters?
   - The **actualparameters** can be passed in the form of **constants** or **variables** or **expressions** to the **formalparameters** which are of **value** type.
   - The **actualparameters** can be passed only as variables to **formalparameters** of **reference** type.

5. Give the General Syntax of a Function Prototype with an Example?
   **Syntax:**
   Datatype space function_name(datatype arguments);
   **Example:**
   ```
   void add(int a, int b);
   void sub(int c, int d);
   ```

6. How the functions that have arguments are invoked in the C++ Program?
   - The functions that have arguments are invoked by
   - **Call by Value Method:**
     - In this method, any change in the formal parameter is not reflected to the actual parameter.
   - **Call by Reference Method:**
     - In this method, any change in the formal parameter is reflected to the actual parameter.

7. Write a note on Default Arguments?
   - The default value is given in the form of variable initialization.
   - The default arguments allow the function call statement with partial or no arguments.
8. **Distinguish between Call by Value and Call by Reference?**

<table>
<thead>
<tr>
<th>CALL BY VALUE</th>
<th>CALL BY REFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>In this method, any change in the formal parameter is <strong>not reflected</strong> to the actual parameter.</td>
<td>In this method, any change in the formal parameter is <strong>reflected</strong> to the actual parameter.</td>
</tr>
<tr>
<td>In this method, the function creates its own <strong>copy</strong> of arguments and then uses them.</td>
<td>In this method, the function works with its own copy of arguments and actually working on original data.</td>
</tr>
<tr>
<td>In this method, the called function arguments, formal parameters create <strong>new variables</strong> to the actual parameters.</td>
<td>In this method, the called function arguments, formal parameters become <strong>aliases</strong> to the actual parameters.</td>
</tr>
</tbody>
</table>

9. **Define Inline Function**

- An **inline function** looks like a normal function in the **source file** but inserts the function **coded directly** into the calling program.
- **Inline function** executes **faster** but requires more **memory space**.

---

**CHAPTER – 5**

**STRUCTURED DATA TYPE – ARRAYS**

1. **Define Array**

   - An **array** is a collection of variables of the same type that are referenced by a **common name**.
   - Array is of two types. They are, One Dimensional and Multi – Dimensional.

2. **What are the types of Array?**

   - Array is of two types. They are, One Dimensional and Multi – Dimensional.
   - **One dimensional Array:**
     - It comprises of finite homogenous elements.
     - Eg: `inta[3];`
   - **Multi – dimensional:**
     - It comprises of elements, each of which is itself a one-dimensional array.
     - Eg: `inta[3][3];`
3. What is One Dimensional Array?
   ∗ One dimensional Array:
     
     It comprises of finite homogenous elements.

   Syntax:
     
     Data type → Space → Array Identifier → [Size];

   Eg:
     
     int a[3];

4. What are the methods to display the contents of the String?
   ∗ There are two methods to display the contents of string. They are,
     ∗ cout method, and cout with write member function
       
       cout<< name - this is similar to any other variable.
       cout.write (pincode, 7); or cout.write (pincode, size of (pincode));

5. Define Write() member function
   ∗ write ( ) is a member function of standard output stream, i.e., ostream.
   ∗ The two parameters required for write ( ) function are identifier string characters, and no. of characters to be displayed.
   ∗ cout.write (pincode, 7); or cout.write (pincode, size of (pincode));

6. What are the String Manipulators defined in string.h header file?

<table>
<thead>
<tr>
<th>Function</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>strlen(s1)</td>
<td>It is used to return the length of the string. i.e. Eg: Length of s1</td>
</tr>
<tr>
<td>strcpy(s1,s2)</td>
<td>It is used to copy the source string (s2) to target string (s1)</td>
</tr>
<tr>
<td>strcmp(s1,s2)</td>
<td>It is used to compare the two strings. If s1==s2 it returns the value as 0</td>
</tr>
<tr>
<td></td>
<td>If s1&gt;s2 it returns the value as &gt;0</td>
</tr>
<tr>
<td></td>
<td>If s1&lt;s2 it returns the values as &lt;0</td>
</tr>
</tbody>
</table>

7. What is Two Dimensional Array?
   ∗ A two-dimensional array is an array in which each element is itself an array.
   ∗ The number of elements in a 2-dimensional array is determined by multiplying the number of rows with number of columns.
     
     Eg: int a[3][3];
8. How the 2-D array is declared? Give with an example?

☆ The declaration of 2-Dimensional Array as follows:

Syntax:

Data type space array_name[row size][column size];

Eg:

int a[3][3];

[Here, in the above example says that a is an array name which has 3 rows and 3 columns.]

[The number of elements in a 2-dimensional array is determined by multiplying the number of rows with number of columns.]

9. Define Matrix

☆ A **Matrix** is a set of **mn** numbers arranged in the form of a **rectangular** array of m rows and n columns.

☆ **Matrices** can be represented through 2-D **arrays**.

Eg:

int a[3][3];

[Here, in the above example says that ais an array name which has 3 rows and 3 columns.]
CHAPTER – 6

CLASSES AND ITS OBJECTS

1. Define Class
   - A class is a way to bind the data and its associated functions together.
   - A class is a new way of creating and implementing a user defined data type.

   **Eg:**
   ```
   class student {
   int a,b,c;
   float d,e,f;
   void add();
   };
   ```

2. What are the specifications of Class?
   A class specification has two parts:
   - Class Declaration
   - Class Function Definitions

3. Write the General Format of the Class?
   **General Format:**
   ```
   class class-name {
   private:
   data members declaration;
   method declaration;
   protected:
   data members declaration;
   method declaration;
   public:
   data members declaration;
   method declaration;
   };
   ```
   **Example:**
   ```
   class ctk {
   private:
   int a,b,c;
   float d,e,f;
   protected:
   void add();
   void sub();
   public:
   void mul();
   void display();
   };
   ```
4. **What are the access specifiers used in the class?**

   The Access specifiers used in the class are,

<table>
<thead>
<tr>
<th>Access Specifiers</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private</td>
<td>Data Members and Member Functions can be accessed within the class.</td>
</tr>
<tr>
<td>Protected</td>
<td>Data Members and Member Functions can be accessed within the class and to the inherited class.</td>
</tr>
<tr>
<td>Public</td>
<td>Data Members and Member Functions can be accessed outside the class.</td>
</tr>
</tbody>
</table>

5. **Define Data Hiding**

   - The members and functions declared under private are not accessible by members outside the class, this is referred to as **data hiding**.
   - **Data Hiding** is the key feature of OOPS.

6. **Define Data Abstraction**

   - Instruments allowing only selected access of components to objects and to members of other classes is called as **Data Abstraction**.
   - **Data abstraction** is achieved through data hiding.

7. **Define Data Member and Member Functions**

   - Data members are the data variables that represent the features or properties of a class.
   - Data members are also called as **attributes**.
   - Member functions are the functions that perform specific tasks in a class.
   - Member functions are called as **methods**.

8. **How is the class members accessed?**

   The class members are accessed using Dot Operator.

   **Eg:**
   ```
   class a
   {
   __________
   void main()
   {
   __________
   a a1; //a1 is the Object for the Class a
   __________
   a1.add();
   __________
   }
   }
   ```

9. **List some characteristics of member function?**

   - Member functions can access the private data of a class. A non-member function cannot do so.
A member function can call another member function directly, without using the dot operator. This is called as nesting of member functions.

The member functions can receive arguments of a valid C++ data type. Objects can also be passed as arguments.

The return type of a member function can be of object data type.

Member functions can be of static type.

10. Define Static Data Member

- Is initialized to zero, only when the first object of its class is created. No other initialization is permitted.
- Only one copy of the member variable is created (as part of the common pool) and is shared by all the other objects of its class type.
- Its scope or visibility is within the class but its lifetime is the lifetime of the program.

CHAPTER 7
POLYMORPHISM

1. Define Function Overloading

- The ability of the function to process the message or data in more than one form is called as Function Overloading.
- In the below example the function area( ) is overloaded.

Example:
void area(float r);
void area(float h, float b, float h1);
void area(float l, float b);

2. How are functions invoked in Function Overloading?

- The compiler adopts BEST MATCH strategy. As per this strategy, the compiler will
  - Look for the exact match of a function prototype with that of a function call statement
  - In case an exact match is not available, it looks for the next nearest match. That is, the compiler will promote integral data promotions and then match the call statement with function prototype.
3. What is Integral Promotions? (CODE: CIFD)
   - char data type can be converted to integer/float/double
   - int data type can be converted to float/double/char
   - float data type to double/char/integer
   - double data type to integer/float

4. What are the Rules for Function Overloading?
   - Each overloaded function must differ either by the number of its formal parameters or their data types
   - The return type of overloaded functions may or may not be the same data type
   - The default arguments of overloaded functions are not considered by the C++ compiler as part of the parameter list
   - Do not use the same function name for two unrelated functions

5. Define Operator Overloading
   - The mechanism of giving special meaning to an operator is called as Operator Overloading.

6. List out the operators that cannot be overloaded?
   - The Operators that cannot be overloaded are,
     - Size of Operator,
     - Conditional Operator,
     - Scope Resolution Operator, &
     - Membership Operator.

7. What are the processes of Operator Overloading?
   - Create a class that defines the data type that is to be used in the overloading operations
   - Declare the operator function `operator()` in the public part of the class.
   - Define the operator function to implement the required operations.

CHAPTER 8
CONSTRUCTORS AND DESTRUCTORS

1. Define Constructor
   - When an instance of a class comes into scope, a special function called the Constructor gets executed.
The Constructor function initializes the class object.
- The memory space is allocated to an object
- The class name and constructor function name is same.
- The Constructor returns nothing.
- It has no data type.

2. **What are the Functions of Constructor?**
   - The constructor function initializes the class object
   - The memory space is allocated to an object

3. **Define Default Constructor**
   - The constructor without parameters is called as Non – Parameterized Constructor.
   - It is also called as Default Constructor.
   - **Eg:**
     ```
     class a
     {
     public:
        a() // Default Constructor
        {
        }
     }
     ```

4. **What is Copy Constructor?**
   - When an object is passed as a parameter to any of the member functions
   - When a member function returns an object
   - When an object is passed by reference to constructor

5. **What are the Rules for Destructor? (Any 4 Points)**
   - The Destructor has the same name as that of the class prefixed by the tilde character (‘~’).
   - The Destructor cannot have arguments.
   - It returns nothing.
   - Destructors cannot be overloaded i.e., there can be only one destructor in a class.
In the absence of user defined destructor, it is generated by the compiler.
The destructor is executed automatically when the control reaches the end of class scope.

6. What are the Rules for Constructor? (Any 4 Points)
   - The name of the constructor must be same as that of the class name.
   - A constructor can have parameter list.
   - The constructor function can be overloaded.
   - The compiler generates a constructor, in the absence of a user defined constructor.
   - It has no data type.
   - The constructor is executed automatically.

7. Define Destructor
   - A destructor is a function that removes the memory of an object which was allocated by the constructor at the time of creating an object.
   - It carries the same name as the class name, but prefixed with tilde (~) character.
   - It returns nothing.
   - It has no data type.

CHAPTER – 9
INHERITANCE

1. Define Inheritance
   - Inheritance is the most powerful feature of an Object Oriented Programming language. (OOP)
   - It is a process of creating new classes called Derived classes, from the existing or Base classes.
   - The derived class inherits all the properties of the Base class.
   - Derived class is a power packed class, as it can add additional attributes and methods and thus enhance its functionality.
   - Example: Vehicle is the Base Class and the Car is the Derived Class

2. What are the Advantages of Inheritance?
   - Reusability of Code – Code developed for one application can be reused in another application.
   - Code Sharing – The Methods of the Base class can be shared by the derived class.
   - Consistency of Interface – The inherited attributes and methods provide a similar interface.
3. Define Derived Class and Base Class
   - **Base Class**
     - A base class is a class from which other classes are derived.
   - **Derived Class**
     - A derived class can inherit members of a base class.

4. What is the syntax of defining the Derived Class?
   **Syntax:**
   ```cpp
   class derived_classname : visibility mode base_classname
   {
   data members of the derived class;
   member functions of derived class;
   };
   
   Eg:
   ```
   ```cpp
   class a // base class
   {
   }
   class b:public a // derived class
   {
   }
   ```

5. Define Accessibility
   - When a member of a base class can be used by the objects or the members of the derived class. This is called as **Accessibility**.
   - The three access specifiers are **private, protected and public**.

6. Define Abstract Class
   - Classes used only for deriving other classes are called as **Abstract Classes**.
CHAPTER – 10
IMPACT OF COMPUTERS ON SOCIETY

1. What are the technical elements we need to reach out the benefits of IT?
   - Connectivity [Computer networks and Internet facility]
   - Affordable computers or other similar devices, and
   - Software

2. Define ATM
   - ATM means Automated Teller Machine.
   - ATM enables withdrawal of money from the accounts in a particular bank anytime and anywhere.

3. Define E-Banking
   - E-Banking permits banking from the comfort of the home by using internet facilities.
   - It has truly improved the reach and services of banks.

4. List some of the areas where software has been developed? (Any 5)
   - Agricultural Finances and Accounting
   - Alternative farming techniques
   - Animal Husbandry
   - Buildings and Irrigation
   - Chat with other agriculturists and scientists
   - Farmland Assessment
   - Fertilizer Analysis
   - Finding links to farm resources, chat boards, classified advertisements, and other farm-related information
   - Gardening
   - Improving Cow Herds and Increasing revenues
   - Land Management
   - Livestock
   - Milk production
   - Use of satellite imagery to decide on the crops
5. What are the areas where computers are used in the HealthCare?

- Computers are used in many areas of healthcare including
  - Hospital Management System
  - Patient Tracking System
  - Exchange of diagnostic records between healthcare units
  - Tracking and Monitoring Communicable Diseases

6. Define E-Learning

- E-Learning that enables online educational programs leading to degrees and certifications

  Eg: Computer Based Tutorials (CBT), and Web Based Tutorials (WBT).

CHAPTER – 11

IT ENABLED SERVICES

1. List some of the IT Enabled Services? (Any 4 Services)

- E-Governance
- Call Centers
- Data Management
- Medical [Telemedicine and Transcription].
- Data Digitization
- Website Services

2. Define Call Center

- A Call Center is sometimes defined as a telephone based shared service center or specific customer activities.
- It is used for number of customer-related functions like marketing, selling, information transfer, advice, technical support and so on.

3. Define Data Management

- Data Management is a category of IT Enabled Services.
- It is used for pertaining to collection, digitization and processing of data coming from various sources.
4. **What is Medical Transcription?**
   - Medical Transcription is a permanent, legal document that formally states the result of a medical investigation.
   - It facilitates communication and supports the insurance claims.
   - There are three main steps involved in Medical Transcription.

5. **Name some of the Application areas where Digitization is used? (Any 5)**
   - Books
   - Database archiving
   - Electronic Catalogues & Brochures
   - Engineering and Design
   - Geographical Information System.
   - Movies, Sounds and High quality image preservation
   - Product/Service Training Manuals
   - Research Journals and Conference Papers

6. **List some of the Organizations that can potentially benefit from the ITES? (Any 5)**
   - Banking
   - Government agencies
   - Hospital
   - Insurance
   - Legal
   - Manufacturing
   - Municipalities
   - Police departments
   - Public utilities
   - Publishing
   - Telecom
   - Transportation

7. **List some website services?**
   - Agriculture Marketing Network
   - Career guidance
   - Employment Online
8. List some Key Benefits of Data Digitization? (Any 4)
   - Long term preservation of the documents.
   - Storage of important documents at one place.
   - Easy to use and access to the information.
   - Quick and focused search of relevant information in terms of images and text.
   - Easy transfer of information in terms of images and text.
   - Easy transfer of information through CD-ROM, internet and other electronic media.

9. What is Digitization?
   - Digitization refers to the conversion of non-digital material to digital form.
   - A wide variety of materials as diverse as maps, manuscripts, moving images and sound may be digitized.

CHAPTER – 12
COMPUTER ETHICS

1. What are the general guidelines on Computer Ethics?
   - Protection of personal data
   - Computer Crime
   - Cracking

2. What are the types of Data Security?
   - Physical Security
   - Personal Security
   - Personnel Security

3. Define Computer Crime
   - A Computer Crime is any illegal activity using computer software, data or access as the object, subject or instrument of the crime.
   - Some Common Crimes are,
     - Virus
4. Define Virus
   - A **Virus** is a self-replicating program that can cause damage to data and files stored on your computer.
   - **57000** known virus programs are in existence.
   - **6** new viruses are found each day.

5. List some of the Computer Crimes? (Any 4 Crimes)
   - Crimes related to money transfer on the internet
   - Making long distance calls illegally using computers
   - Illegal access to confidential files
   - Stealing hardware
   - Selling or misusing personal
   - Hardware and software piracy
   - **Virus**
   - **Cracking**
   - **Theft of computer time**

6. Define Piracy
   - Making and using duplicate hardware and software is called **piracy**.

7. What is meant by Theft of Computer Time?
   - Some other software runs on an idle computer without the knowledge of the organization. This is called “**Theft of Computer Time**”.

8. Define Cracking
   - **Cracking** is the illegal access to the network or computer system.
   - Illegal use of special resources in the system is the key reason for cracking.
   - The resources may be hardware, software, files or system information.