

OBJECT ORIENTED PROGRAMMING

UNIT – I

PART A

1. Define object oriented programming?

OOP is an approach that provides a way of modularizing programs by creating partitioned memory areas for both data and functions that can be used as templates for creating copies of such modules on demand.

2. List some features of OOP?

- i. Emphasis is on data rather than procedures.
- ii. Programs that are divided into what are known as objects.
- iii. Follows bottom – up approach in program design.
- iv. Functions that operate on the data of an object are tried together in the data structure.

3. Define Class?

- A Class is a collection of objects of similar type.
- The classes are user-defined data types and behave like built-in types of a programming language.
- A class is a way to bind the data and its associated functions together.
- A class is a user-defined data type with a template that serves to define its properties.
- A class is a blueprint that defines the variables & the methods common to all objects of a certain kind.

4. What do you mean by object?

Objects are basic run-time entities in an object-oriented system. They may represent a person, a place, a bank account, a table of data or any item that the program has to handle. Each object has the data and code to manipulate the data and these objects interact with each other.

5. What is meant by Encapsulation?

The wrapping up of data and function into a single unit(class) is known as Encapsulation.

6. What do you mean by Data abstraction?

Abstraction refers to the act of representation of essential features without including the background details or explanations. Classes use the concept of abstraction & are defined as a list of abstraction attributes such as size, weight & cost & functions to operate on these attributes.

7. What do you mean by inheritance?

Inheritance is the process by which objects of one class acquire the properties of objects of another class.

8. What do you mean by reusability?

The process of adding additional features to an existing class without modifying it is known as „Reusability“. The reusability is achieved through inheritance. This is possible by deriving a new class from an existing class. The new class will have the combined features of both the classes.

9. Define polymorphism?

Polymorphism means the ability to take more than one form. For example, an operation may exhibit different behavior in different instances. The behavior depends upon the types of data used in the operation.

Example: Consider the operation of addition.

For two numbers, the operation will generate a sum.

For two strings, the operation will generate a concatenation.

- There are two types of polymorphism: Compile time polymorphism and Run time polymorphism

10. Define Compile time polymorphism / Early Binding / static Binding

Compile time polymorphism is implemented by using the overloaded functions and overloaded operators.

The overloaded operators or functions are selected for invocation by matching arguments both by type and number. This information is known to the compiler at the compile time therefore the compiler is able to select the appropriate function for a particular function call at the compile time itself. This is called as „Early Binding“ or „Static Binding“ or „Static Linking“ or „Compile time polymorphism“. Early binding simply means an object is bound to its function call at the compile time. At runtime, when it is known what class objects are under consideration, the appropriate version of the function is invoked. Since the function is linked with a particular class much later after its compilation, this process is termed as „late binding“. It is also known as dynamic binding because the selection of the appropriate function is done dynamically at run time. This runtime polymorphism can be achieved by the use of pointers to objects and virtual functions.

12. What do you mean by message passing?

Objects communicate with one another by sending and receiving information. A message for an object is a request for execution of a procedure, and therefore will invoke a function in the receiving object that generates the desired result. Message passing involves specifying the name of the object, the name of the function and the information to be sent.

13. List out the benefits of OOPS.

- Through inheritance, we can eliminate redundant code and extend the use of existing classes.
- The principle of data hiding helps the programmer to build secure programs.
- It is possible to have multiple instances of an object to co-exist without any interference.
- Object oriented systems can be easily upgraded from small to large systems.
- Software complexity can be easily managed.

14. List out the applications of OOPS.

- Real time systems
- Simulation and modeling
- Object oriented data bases
- AI and expert systems

14. What is an expression?

An expression is a combination of operators, constants and variables arranged as per rules of the language. It may include function calls which return values.

16. Define keywords?

Keywords are explicitly reserved identifiers and cannot be used as names for the program variables or other user defined program elements.

17. Why do we need the preprocessor directive #include <iostream.h>?

This directive causes the preprocessor to add the contents of the iostream.h file to the program. It contains the declarations for the identifier cout and the operator <<. It contains function prototypes for the standard input output functions.

18. What is the use of return statement in main() function?

In C++, main() returns an integer type value to the operating system. Therefore, every main() in C++ should end with a return(0) statement; otherwise a warning or an error might occur.

19. How does a main() function in C++ differ from main() in C?

In C++, main() returns an integer type value to the operating system but in C, main() returns nothing to operating system by default.

20. What is formal parameter?

If a function is to use arguments, it must declare variables that will accept the values of the arguments. These variables are called the formal parameters of the function.

Example : int max(int a, int b) // Variables a and b are formal parameters {
if(a>b) return a; return b; }

21. What is global variable?

Global variables are known throughout the program and may be used by any piece of code. Also, they will hold their value throughout the program's execution.

22. What is the use of exit() function?

The exit() function causes immediate termination of the entire program, forcing a return to the operating system.

The general form :

Void exit(int return code);

The value of return code is returned to the calling process, which is usually the operating system. Zero is generally used as a return code to indicate normal program termination.

23. What is the use of break and continue statements?

Break is used to terminate a case in the switch statement. Force immediate termination of a loop, bypassing the normal loop conditional test.

Continue is used to force the next iteration of the loop to take place, skipping any code in between.

24. What is array? How do you declare Single dimensional array in C++?

An array is a collection of variable of the same type that are referred to through a common name. A specific element in an array is accessed by an index.

General form for declaring one dimensional array:

`type var-name[size];`

Here type declares the base type of the array and size defines how many elements the array will hold. Example: `int x[10];` // elements are `x[0]` ,`x[1]``x[9]`

25. What is two dimensional array?

It is an array of one dimensional arrays. It has two subscripts.

General form for declaring two dimensional array:

`Data type var-name [size1][size2];`

Here size1 refers row index and size2 refers to column index. The number of elements in the array is `size1*size2`.

Example:

`int x[5][10];`

Here x is the two dimensional array having 5 rows and 10 elements in each row. Number of elements in the array is $5*10=50$ elements.

PART B

1. Basic concepts of oops.
2. What is Object oriented Paradigm? Explain the various Features of oops
3. Advantages of OOP.
4. Write about Merits &Demerits of OOP.
5. Explain object oriented languages?
6. What are the applications of OOPs?
7. Write about If. Else Statements and Switch Statement and Do...While Statement.
8. Write about functions in C++ in detail.

UNIT – II

PART A

1. What are pointers?

A pointer is a variable that holds a memory address. This address is the location of another object in memory.

2. What are pointer operators?

The pointer operators are * and &.

The & is a unary operator that returns the memory address of its operand.

The * is a unary operator that returns the value located at the address that follows.

3. What is meant by storage class specifiers?

Storage class specifiers tell the compiler how to store the subsequent variable.

There are five storage class specifiers supported by C++:

i. extern ii. static iii.register iv. auto v.mutable

4. What is the use of 'extern' variables?

In a multifile program, we can declare all of the global variables in one file and use extern declarations in the other without defining it again.

The extern keyword has this general form:

extern var-list;

5. Write about union.

A union is a memory location that is shared by two or more different variables, generally of different types, at different times. Declaring a union is similar to declaring a structure. General form is:

union union-type-name{tmember-name; member-name;.....} union-variables;

It allocates enough storage to hold the largest member of the union.

6. What is enumerated data type?

An enumerated data type is a user defined data type which provides a way of attaching names to numbers.

Ex., enum shape{circle,square,rectangle};

enum color{red,blue,green};

Now the tag names shape and color become new type names. We can declare new variables using them.

7. what is function?

Functions are the building blocks of C++ and the place where all program activity occurs. The general form is

ret-type function-name(parameter list)

{body of the function }

The ret-type specifies the type of data that the function returns. The parameter list is a comma-separated list of variable names and their associated types that receive the values of the arguments when the function is called. When the function is called the control is transferred to the first statement in the body.

8. What are the two ways to pass arguments to the function?

Call by value: This method copies the value of an argument into the formal parameter of the function.

Call by reference: This method copies the address of an argument into the formal parameter of the function.

9. How to create call by reference?

We can create call by reference by passing a pointer (i.e. address of the argument) to an argument, instead of argument itself.

Example:

```
void swap (int *x,int *y)
{
    int temp;
    temp=*x;
    *x=*y;
    *y=temp;
}
```

this function can be invoked with the addresses of the arguments
as swap(&i,&j); //for interchanging the integer values i and j

10. What is function prototype?

In C++ all functions must be declared before they are used. This is normally accomplished using function prototype. Using function prototype, the compiler can find and report any illegal type conversions between the type of arguments used to call a function and the type definition of its parameter.

General form is: ret-type function-name(type param-name1,type param-name2,.....type param-namen);

11. What is the difference between endl and '\n'?

Using endl flushes the output buffer after sending a '\n', which means endl is more expensive in performance. Obviously if you need to flush the buffer after sending a '\n', then use endl; but if you don't need to flush the buffer, the code will run faster if you use '\n'.

12. What is the use of reference variables?

A reference variable provides an alias (alternate name) for a previously defined variable. A reference variable is created as follows:

Datatype & reference-name =variablename;

Example: int i=10;

```
int &sum=i;
cout<<sum; // output:10
sum=100;
cout<<i; // outp
```

16. How objects are created?

Once a class has been declared, we can create variables of that type by using the class name .The class variables are known as objects .The necessary memory space is allocated to an object at the time of declaration.

17. How the members of a class can be accessed?

The private data of a class can be accessed only through the member functions of that class. The format for calling a member function:

Objectname.function_name(actual _arguments);

18. What are the two ways of defining member functions?

Member functions can be defined in two places

- Outside the class definition
- Inside the class definition

14. What is the general form of a class definition in C ++

```
Class class_name
{
    Private:
        Variable declaration;
        Function declaration;
    Public:
        Variable declarations;
        Function declarations;
};
```

15. Differentiate Structure and Classes

Structures	Classes
1. By default the members of the structure are public.	1. By default the members of Class are private.
2. Data hiding is not possible	2. Data hiding is possible
3. Structure data type cannot be treated like built-in-type	3. Objects can be treated like built-in-types by means of operator overloading.
4. Structures are used only for holding data	4. Classes are used to hold data and functions
5. Keyword „struct“	5. Keyword „class“

19. What do you mean by nesting of member functions?

A member function can be called by using its name inside another member function of the same class. This is known as nesting of member functions.

20. Explain about 'static' variables.

The special characteristics of static member variable are

- It is initialized to zero when the first object of its class is created.
- Only one copy of that member is created for the entire class, no matter how many objects are created.
- It is visible only within the class, but its lifetime is the entire program

21. Explain about static member functions.

- A member function can have access to only other static members declared in the same class

- A static member function can be called using the class name as follows

Classname :: function name;

22. What are the ways of passing objects as function arguments?

Pass by value : A copy of the entire object is passed to the function. Any changes made to the object inside the function don't affect the objects used to call the function.

Pass by Reference: Only the address of the object is transferred to the function. When an address of the object is passed, the called function works directly on the actual object used in the call. This means that any changes made to the object inside the function will reflect in the actual object.

23. What do you mean by friend function?

By declaring a non member function as friend of a class, we can give full rights to access its private data members (i.e.) A friend function although not a member function has full access rights to the private members of the class.

24. What are the special characteristics of a friend function?

- It is not in the scope of the class to which it has been declared as friend.
- Since it is not in the scope of the class, it cannot be called using the object of the class.
- It can be invoked like a normal function without the help of any object
- Unlike member functions it cannot access the member names directly and has to use an object name and dot membership operator with each member name
- Usually it has the object as arguments.

25. What is a const member function?

If a member function does not alter any data in the class, then we may declare it as a const member function.

e.g. : void getbalance () const;
void mul(int,int) const;

26. What is a main function?

All the C++ programs start with the function main(). Function main returns the integer value that indicates whether the program executed successfully.

27. What is the purpose for the return statement?

The return statement is used to return the value from a function. The statement return 0; returns the value 0. The return statement supplies a value from the called function to the calling function.

28. Explain function prototype?

It is used to describe the function interface to the compiler by giving details such as type number and type arguments and the type of return values. Function prototype is a declaration statement in the calling program.

Syntax: Type function_name (arguments);

29. Define macro?

A short piece of text or text template that can be expanded into a longer text.

30. What do you inline function?

A function definition such that each call to the function is in effect replaced by the statements that define the function.

31. What are the situations that inline functions may not work?

1. For function returning values, if a loop, a switch, or a goto exists.
2. For function not returning values, if a return statement exists.
3. If function contains static variables.
4. If inline functions are recursive.

32. What are constant arguments?

The qualifier const tells the compiler that the function should not modify the argument. The compiler will generate an error when this condition is violated.

PART B

- 1.Explain the declaration of a class in c++. How will you define the member function of a class? Explain.
- 2.What is the need for parameterized constructor? Explain the function of constructors with their declaration inside a class
- 3.Explain all types of inheritance.
- 4.What is virtual function? Give an example to highlight its need?
- 5.What is a Friend function? Describe their benefits and limitations? Give Suitable example.
- 6.What is meant by function overloading? Write the rules associated with Function overloading. Give suitable example to support your answer?
- 7.Explain virtual base classes and virtual function, pure virtual function.

UNIT – III

PART A

1. What do you mean by constructor?

- Constructor is a member function
- It initializes the objects of its class
- It has the same name as the class
- It is automatically invoked when the objects are created
- It constructs the value of the data members of the class
- It has no return type.

2. What do you mean by automatic initialization of objects?

C++ provides a special member function called the constructor which enables an object to initialize itself when it is created. This is known as automatic initialization.

3. Define default constructor?

A constructor that accepts no parameters is called default constructor.

Ex : The default constructor for Class A is

A:: A().

4. What is parameterized constructor?

The constructor that take arguments are called parameterized constructor. When a constructor has been parameterized the object declaration statement such as integer i1 will not work.

5. What are the two ways in which a parameterized constructor can be called?

- By calling the constructor explicitly
- By calling the constructor implicitly

6. What are the kinds of constructors that we call?

- Constructor without arguments
- Constructor with arguments

7. What is the advantage of using dynamic initialization?

The advantage of using dynamic initialization is that various initialization formats can be provided using overloaded constructors.

8. What are copy constructors? Explain with example?

The constructor that creates a new class object from an existing object of the Same class. Example: integer i2(i1) or integer i2 = i1 would define the object i2 at the same time initialize the values of i1.

9. What do you mean by constructor overloading?

When more than one constructor function is defined in a class, then it is called constructor overloading.

10. What do you mean by Dynamic constructors?

The constructors can be used to allocate memory while creating objects. This will enable the system to allocate the right amount of memory for each object, thus resulting in saving of memory. Allocation of memory to objects at the time of their construction is known as dynamic construction of objects.

11. What do you mean by destructor?

- Destructor is used to destroy the object that has been created by a constructor.
- It is a member function, whose name is same as the class name preceded by a tilde.
- The destructor never takes any arguments nor does it return any value.
- It will be invoked implicitly by the compiler upon exit from the program to clean up storage. Ex., `~integer () { }`

12. Write some special characteristics of constructor

- They should be declared in the public section
- They are invoked automatically when the objects are created
- They do not have return types, not even void and therefore, and they cannot return values
- They cannot be inherited, though a derived class can call the base class
- They can have default arguments
- Constructors cannot be virtual function

13. List the difference between constructor and destructor?

Constructor can have parameters. There can be more than one constructor. Constructors is invoked when from object is declared.

Destructor has no parameters. Only one destructor is used in class. Destructor is invoked up on exit program.

14. How do you allocate / unallocated an array of things?

Use “`p = new T(n)`” for allocating memory and “`delete () p`” is for releasing of allocated memory. Here p is the array of type T and of size n. Example:

```
Fred*p=new Fred[100]; // allocating 100 Fred objects to p
```

```
...
```

```
delete[]p; //release memory
```

Any time we allocate an array of objects via new , we *must* use [] in the delete statement. This syntax is necessary because there is no syntactic difference between a pointer to a thing and a pointer to an array of things.

15. Can you overload the destructor for class?

No. You can have only one destructor for a class. It's always called `Fred::~~Fred()`. It never takes any parameters, and it never returns anything. You can't pass parameters to the destructor anyway, since you never explicitly call a destructor.

16. What is the advantage of using dynamic initialization?

The advantage of using dynamic initialization is that various initialization formats can be provided using overloaded constructor.

17. Define operator overloading?

A language feature that allows a function or operator to be given more than one definition

18. Give the operator in C++ which cannot be overloaded?

- i. `sizeof` -> size of operator
- ii. `::` -> scope resolution operator
- iii. `?:` -> conditional operator
- iv. `.` -> Membership operator
- v. `.*` -> pointer to member operator

19. How can we overload a function?

With the help of a special operator called operator function. The general form of an operator function is:

```
Return type class name :: operator op(arg list)
{Function body }
```

20. Give any four rules for operator overloading?

- (i) Only existing operators can be overloaded.
- (ii) The overloaded operator must have at least one operand that is of user defined type.
- (iii) We cannot use friend functions to overload certain operators.
- (iv) Overloaded operators follow the syntax rules of the original operators.

21. What are the steps that involve in the process of overloading?

- Create a class that defines the data type that is to be used in the overloading operation.
- Declare the operator function `operator op()` in the public part of a class.
- Define the operator function to implement the required operation.

22. What are the restriction and limitations overloading operators?

Operator function must be member functions or friend functions. The overloading operator must have at least one operand that is of user defined datatype.

23. Give a function overload a unary minus operator using friend function? friend void operator -(space &s)**24. List the difference between constructor and destructor?**

Constructor can have parameters. There can be more than one constructor. Constructors are invoked when an object is declared.

Destructors have no parameters. Only one destructor is used in a class. Destructors are invoked up on exit program.

25. How do you overload a dereferencing operator ->?

The dereferencing operator `->`, also called the class member operator, can be used as a unary postfix operator. The general usage is shown here: `Object->element`;

Here, object is the object that activates the call. The operator `->()` function must return a pointer to an object of the class that operator `->()` operates upon. The element must be some member accessible within the object.

Note: An operator -() function must be a member of the class upon which it works.

26. How do you overload a comma operator?

The comma is a binary operator. We can make an overloaded comma perform any operation we want. If we want the overloaded comma to perform in a fashion similar to its normal operation, then our version must discard the values of all operands except the right most. The right most value becomes the result of the comma operation.

PART B

1. Explain the operator overloading of Unary operators in C++.
2. Explain the operator overloading of Binary operators in C++.
3. How is type conversion implemented in C++?
4. Elaborate upon the inheritance methods and describe with an example.
5. Describe virtual functions with an example.
6. Explain polymorphism?
7. Write a c++ program to overload new operator
8. Write a c++ program to add two matrices using overloaded

PART A

1. Define inheritance?

The mechanism of deriving a new class from an old one is called inheritance. The old class is referred to as the base class and the new one is called the derived class or the subclass.

2.What are the types in inheritance?

- i. Single inheritance
- ii. Multiple inheritance
- iii. Multilevel inheritance
- iv. Hierarchical inheritance
- v. Hybrid inheritance

3. Explain single inheritance?

A derived class with only one base class is called single inheritance

4. What is multiple inheritance?

A derived class with more than one base class is called multiple inheritance.

5. Define hierarchical inheritance?

One class may be inherited by more than one class. This process is known as hierarchical inheritance.

6. What is hybrid inheritance?

There could be situations where we need to apply two or more type of inheritance to design a program. This is called hybrid inheritance.

7. What is multilevel inheritance?

The mechanism of deriving a class from another derived class is known as multilevel inheritance.

8. What is virtual base class?

The child has two direct base classes parent1 and parent2 which themselves have common base class grand parent. The child inherits the traits of grand parent via two separate paths. It can also inherit directly shown by the broken line. The grand parent is sometimes referred to as indirect base class.

9. What is abstract class?

An abstract class is one that is not used to create objects. An abstract class is designed only to act as a base class. It is a design concept in program development and provides a base upon which other classes may be built.

10. What is nesting of classes?

A class can contain objects of other classes as its members. It is called nesting of classes.

```
Class A{....};  
Class B {  
    A a; };
```

11. What is polymorphism?

Polymorphism is one of the crucial feature of OOP. It simply means one name multiple forms. It contains run time polymorphism and compile time polymorphism.

12. What are the types of polymorphism?

- i. Run time polymorphism
- ii. Compile time polymorphism

13. What is compile time polymorphism?

The overloaded member function are selected for invoking by matching arguments both type and number. This information is known to the compiler at the compile time and therefore compiler is able to select the appropriate function for a particular call at the compile time itself. This is called early binding or static binding or static linking. Also known as compile time polymorphism.

14. What the types in compile time polymorphism?

- ^ Function overloading
- ^ Operator overloading

15. What is this pointer?

A unique keyword called this to represent an object has invokes a member function.

16. What is virtual function?

The same function same in both the base and derived classes the function in base class is declared as virtual using the keyword virtual preceding its normal function declaration.

17. What are the rules for virtual function?

1. They cannot be static members
2. They are access by using object pointers
3. A virtual function can be a friend of another class.

18. What is pure virtual function?

A pure virtual function is a function declared in a base class that has no definition relative to the base class.

19. What's the difference between public, private, and protected?

A member (either data member or member function) declared in a private section of a class can only be accessed by member functions and friends of that class

A member (either data member or member function) declared in a protected section of a class can only be accessed by member functions and friends of that class, and by member functions and friends of derived classes

A member (either data member or member function) declared in a public section of a class can be accessed by anyone.

20. Why can't derived class access private things from base class?

To protect from future changes to the base class. Derived classes do not get access to private members of a base class. This effectively "seals off" the derived class from any changes made to the private members of the base class.

21. What are the different types of binding of object with pointer?

When you have a pointer to an object, the object may actually be of a class that is derived from the class of the pointer.

Thus there are two types:

1. Static binding
2. Dynamic binding

22. Define the keyword 'protected'?

A member declared as protected is accessible by the member function within its class and any class immediately derived from it. It cannot be accessed by the functions outside these two classes.

23. Give the visibility of base class members in derived class?

Base Class	Derived Class Visibility		
Visibility	Public derivation	Private derivation	Protected derivation
Private	Not inherited	Not inherited	Not inherited
Protected	Protected	Private	Protected
Public	Public	Private	Protected

24. How ambiguity is resolved in inheritance?

In the inheritance when more than one class has a function with the same name and same type signature, then there exists ambiguity. It can be solved by using the class resolution operation with the function.

25. Define templates

26. The templates is the most sophisticated & high powered feature of C++.
- Using template it is possible to create generic functions and classes. In a generic function or class, the type of data which the function or class operated is specified as a parameter. Thus it is possible to use one function or class with several different types of data without having to explicitly recode versions of each data type.

26. What is Generic function? Or Define function template

A generic function defines a general set of operations that will be applied to various types data. The type data that the function will operate upon is passed to it as a parameter. Through a generic function, a single general procedure can be applied to a wide range of data.

28. Define generic classes?

Using generic classes, we can create a class that defines all the algorithms used by the class. The actual type of data being manipulated will be specified as a parameter when objects of that class are created.

PART B

1. Design an object oriented system for student information system
2. Analyze the reason for popularity of Object Oriented programming. Compare OOP with POP.
3. What do u mean by Operator overloading? What are the rules for overloading operators?
4. Discuss the need for Virtual Destructor with example.
5. Distinguish between early and late binding with examples? Discuss each in detail
6. Explain the kinds of classes with examples for each.
7. Explain about Exception handling with suitable example.
8. Define a class to represent a BANK ACCOUNT . Include the following members.
Name of the depositor, Account number, Type of account, balance amount in the account. To Provide the member function to perform the following tasks.
To assign initial values
To deposit an amount
To withdraw an amount after checking the balance

UNIT – V

1. Define streams

A stream is a logical device that produces or consumes information. A stream is linked to a physical device by the I/O system.

A stream is a sequence of bytes. It acts either as a source from which the input data can be obtained or as a destination to which the O/P data can be sent.

A stream acts as an interface between the program and the I/O devices.

2. What are the two types of I/O operation?

- Formatted I/O operation - width(), precision(), fill(), setf().
- Unformatted I/O operation - cin>>,cout <<,getline(), write().

3. What are the ios format function?

Function	Task
width()	To specify the required field size for displaying an o/p value.
precision()	To specify the number of digits to be displayed after the decimal point of float variable.
fill()	To specify a character that is used to fill the unused portion of a field.
setf()	To specify format flags that can control the form of o/p display (ex: left justification and right justification)
unsetf()	To clear the flags specified

4. What are the unformatted I/O operations we have in C ++ ?

cin >> -to read the data of any type excluding white spaces.

cout<< - to display data of any type.

get() -read a single character form keyboard.

put () - display a single character

getline() -to read a line of text ends with a new line character

write() -to read a line of text

5. What are the features available in C ++ for formatting O/P?

- ios class functions & flags
- manipulators
- User defined output functions

6. What are the advantages & disadvantages of manipulators?

Manipulators are functions that can be used to manipulate the output formats. They provide the same features as that of ios functions . Some manipulators are more convenient to use than their counterparts in the ios class. For example two or more manipulators can be used as a chain in our statements as shown

cout<< manip1<<manip2<<manip3<<item;

ios member function return the previous format state which can be used later if necessary. But the manipulators does not return to the previous format state.

7. Define file

File is a collection of related data stored in a particular area on the disk.

8. What are the two kinds of data communication a program may encounter?

- Data transfer between the console unit and the program
- Data transfer between the program and disk file.

9. What are the two methods of opening a file?

- Using the constructor function of the class - used when only one file in a stream
- Using the member function `open()` of the class- used when we want to manage multiple files using one stream.

10. Describe how to open a file using constructor

1. Create a file stream object to manage the stream using appropriate class. The class `ofstream` is used to create the output stream & `ifstream` is used to create the input stream.
2. Initialising the file object with the desired filename.
e.g ., `ofstream outf("results");// output only`
`ifstream inf("data");//input only`

11. Describe how to open files using `open()` function

`Open()` can be used to open multiple files that use the same stream object.

If we want to process a set of files sequentially then we create a single stream object and use it to open each file in turn.

General Format:

```
file_stream_class stream_object;  
stream_object.open("filename");
```

Ex., `ofstream outf;`
`outf.open("data");//for output`
`outf.close();//disconnect stream from data1`
`outf.open("data2");//connect stream to data2`
`outf.close(); //disconnect stream form data2`

12. What is the use of passing second parameter to `open()` function? What are they?

The second argument mode (file mode parameter) specifies the purpose for which the file is opened. The file mode parameter can take one of constants defined in the class `ios`.

Parameter	Meaning
<code>ios::app</code>	Append the end of file
<code>ios::ate</code>	Go to end of file for opening
<code>ios::binary</code>	Binary file
<code>ios::in</code>	Open file for reading only
<code>ios::nocreate</code>	Open fails if the file does not exist
<code>ios::noreplace</code>	Open files if the file already exists
<code>ios::out</code>	Open file foe writing only
<code>ios::trunc</code>	delete the contents of the file if it exists

13. What is the use of file pointers?

The input pointer is used for reading the contents of a given file location and the output pointer is used for writing to a given location . Each time an input or output operation takes place, the appropriate pointer is automatically advanced.

14. What are the errors we may expect while dealing with the files?

1. A file which we attempting to open for reading does not exist
2. The file name used for a new file may already exist
3. We may attempt an invalid operation such as reading past the end-of-file.
4. There may not be any space in the disk for storing more data
5. We may use an invalid filename
6. We may attempt to perform an operation when the file is not opened for that purpose

15. What are the error handling functions for files available in C ++?eof()

1. fail()
2. bad()
3. good()

16. What are the default actions on file pointers while opening a file?

When a file is opened in read only mode, the input pointer is automatically set at the beginning so that the file can be read from the start.

Similarly when a file is opened in write only mode the existing contents are deleted and the output pointer is set at the beginning. This enables to write to the file from the beginning. If the file is opened in append mode the output pointer is set at the end of the file.

17.What are the functions in C ++ to manipulate file pointers ?

- seekg() - moves get pointer (i/p) to a specified location.
- seekp() - moves put pointer(o/p) to a specified location.
- tellg() - gives the current position of the get pointer.
- tellp() - gives the current position of the put pointer.

18. What are the file stream classes in C++?

1. filebuf
2. ifstream
3. ofstream
4. fstream

19. What are the file manipulation function in C++?

- i. seekg()
- ii. seekp()
- iii. tellg()
- iv. tellp()

20. What are the file open modes?

ios::app, ios::binary, ios::out, ios::in, ios::ate, ios::nocreate, ios::noreplace, ios::trunk.

21. What are the error handling function in C++?

1. eof()
2. fail()
3. bad()
4. good()

22. What is meant by exception handling?

C++ provides a built-in error handling mechanism called Exception handling. Exception handling is the subsystem of C++ that allows us to handle errors that occur at run time in a structured and controlled manner. With exception handling, program can automatically invoke an error handling routine when an error occurs. The proper use of exception handling helps to create resilient code.

23. What are two types of exceptions?

Exceptions are of two kinds namely

Synchronous : Error such as “out-of-range index” and “overflow”

Asynchronous : errors that are caused by events beyond the control of the program such as keyboard interrupts.

24. What are the tasks to be handled by exception handling?

1. Find the problem(hit the exception)
2. Inform that an error has occurred(throw the exception)
3. Receive the error information(catch the exception)
4. Take corrective actions(handle the exception) General format:

```
try
{throw exception;//block of statements which
    //detects &throws exception }
catch(type arg)//catches exception
{-----//block of statements
    -----//that handles the exception }
```

25. How to handle all types of exception?

In some situations, it is not possible to anticipate all possible types of exceptions and therefore may not possible to design independent catch handlers to catch them. In such circumstances, a catch statement can be forced to catch all exceptions instead of certain types alone. This is achieved by defining the catch statement using

```
catch(...)
{ statements for processing all exceptions
}
```

26. Explain exception handling mechanism.

When the try block throws an exception, the program control leaves the try block and enters the catch statement of the catch block. The exceptions are object used to transmit information about a problem. If the type of the object thrown matches the argument type in the catch statement, then the catch block is executed for handling exception. If they do not match the program is terminated using abort () function.

PART B

P

1. Write a c++ program to perform multiplication between an integer and complex number object using friend operator function.
2. Write functions in c++ using function overloading to compute the area of a square, circle, triangle, rectangle
3. Define a class String to represent the value of the string and its length. Allocate memory dynamically for the string. Overload the following operators as + for concatenation [] extract the characters at specified index () length of the string Write a c++ program to test this class.
4. Create class called shopping list . The shopping list includes details such as the item code number and price of item. Provide operations such as Adding an item to the list Deleting an item from the list Printing the total value .
5. Design a base class called shape . Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes triangle and rectangle from the base class shape. Add to the base class a member function getdata() to initialize base class members and another member function display area() to compute and display the area of figures
6. Write a program in c++ to implement set with overloaded operators +, -, < , > for the operations of union, intersection, subset and superset of two sets, respectively.
7. Write an object oriented program in c++ to list all the palindromes in a given text.
8. Write a c++ program to display the given number in words for example. If the input is 123 and the output is ONE TWO THREE.