1. What is a class?

Class is a collection of related objects. It’s a user defined data type which consists of any combination of different data types and their related functions.

1. What are literals?

Literals are expressions with a fixed value. The data items whose value never changes during execution of the program are called literals.

1. What are constants?

A constant is avalue of any type that has the same value and never change.

1. Difference between While and do while loop.

**While**  **Do loop**

It is an entry controlled loop Exit controlled loop

It the condition is true then only it It is executed atleast once.

executes the body of the loop

1. What is Function Overloading?

The concept of permitting may functions to have the same name but different function signature. Signature means that the number of arguments, or the type of argument or both should be different.

1. What is Polymorphism?

Polymorphism allows a client to treat different objects in the same way even if they were created from different classes and exhibit different behaviors. You can use implementation inheritance to achieve polymorphism in languages such as C++ and Java. Base class object's pointer can invoke methods in derived class objects. You can also achieve polymorphism in C++ by function overloading and operator overloading.

1. Differentiate between Global and local variable.

Global variable Local Variable

It is a variable which is declared outside It is a variable which is declared within a function statement a function or within a compound

It is accessible throughout the program. It is accessible only within the function.

1. Characteristics of Destructors.

It has a name that of the class. Destructors are called automatically after the end of the program or block. They don’t have a return type. They cannot be overloaded. It is declared with the ~sign.

1. Differentiate between Call by value and Call by Reference in Functions.

Call by value Call by reference

creates copy of the data of the actual is used to share the same memory parameter and formal parameter location for the actual and formal

parameter.

The changes done in the function in formal The changes done in the function are

Parameter are not reflected back in the reflected back in the calling

calling environment. Environment.

1. What is the difference between OOPs and Procedural programming

Object Oriented programming Procedural programming

Emphasis on data. Emphasis on doing things(functions)

Follows bottom-up approach follows top down approach in program design

Data hiding feature prevents accidental presence of global variable increases

change in data chances of change in data

1. What are actual parameters?

Is a parameter, which is used in function call to send the value to function header parameters .

1. What are formal parameter?

It is a parameter, which is used in function header, to receive the value from the calling environment.

1. What are Private members of the class?

Private members of the class can be accessed in the entire class but non- essential details of the object which are hidden from the user.

1. What are Public members of the class

Public members of the class has file scope they can be accessed in the entire program.

1. What are objects ?

Real time entities with properties are called objects. Objects are instances of the class.

Constructors and Destructors

Constructor:- A constructor is a special member function with the same name as its class name and

is used to initialize the data members of the class. Constructor is invoked automatically when an

object of a class is created. Constructor do not return any value not even void. Constructor must be

defined in public section.

Types of Constructors

1. Default Constructor (No argument constructor):- A default constructor accepts no

parameters. When no constructor is defined in the class, compiler provides the default

constructor.

2. Parameterized Constructor (Overloaded Constructor):- Parameterized constructor

accepts parameters and is used to assign these parameters to the data members of the class.

There may be many definitions of the parameterized constructor depending upon the type and

number of parameters passed to the constructor and so it is also called overloaded

constructor.

3. Copy Constructor:-A constructor that accepts a reference to an instance of its own class as

an argument is called as Copy Constructor. A copy constructor is used to create new object

with the similar values of existing object. A copy constructor is invoked when one object is

defined and initialized with another object of the same class.

**What is a copy constructor? Which situation is it invoked in?**

The copy constructor is a constructor which creates an object by initializing it with an object of the same class, which has been created previously. The copy constructor is used to:

* Initialize one object from another of the same type.
* Copy an object to pass it as an argument to a function.
* Copy an object to return it from a function.

**1. Data Abstraction**: Abstraction refers to the act of representing essential features without including the

background details or explanations. Abstraction is implemented through public members of a class, as the outside world is given only the essential and necessary information through public members, rest of the things remain hidden.

**2. Data Encapsulation**: The wrapping up of data and operations/functions (that operate o the data) into a single unit (called class) is known as Encapsulation. Encapsulation is implemented with the help of a class as a class binds together data and its associated function under one unit.

**3. Modularity**: The act of partitioning a program into individual components is called modularity. C++ implements modularity through separately compiled files. The traditional practice in the C++ community is to place module interface in files named with a .h suffix; these are called header files which can be used through #include directive.

**4. Inheritance**: Inheritance is the capability of one class of things to inherit capabilities or properties from another class. Inheritance is implemented in C++ by specifying the name of the (base) class from which the class being defined (the derived class) has to inherit from.

**5. Polymorphism**: Polymorphism is the ability for a message or data to be processed in more than one form. C++ implements polymorphism through virtual functions, overloaded functions and overloaded operators.

When will you make a **function Inline** and why?

We will make a function inline when the functions are small and are called often. Inline functions run a little faster than the normal functions as the compiler replaces the function call statement with the function code itself and then compiles the entire code. Thus, with inline functions, the compiler does not have to jump to another location to execute the function, and then jump back as the code of the called function is already available to the calling program.