**Q1. Write basic concepts of OOPS?**

Following are the concepts of OOPS and are as follows:.

1. Encapsulation
2. Abstraction/Data Hiding.
3. Inheritance.
4. Polymorphism.
5. Persistence

**Q2. What is a class?**

A class is simply a representation of a type of object. It is the blueprint/ plan/ template that describe the details of an object.

**Q3. What is an object?**

Object is termed as an instance of a class, and it has its own state, behavior and identity.

**Q4. What is Encapsulation?**

Encapsulation is an attribute of an object, and it contains all data which is hidden. That hidden data can be restricted to the members of that class.

Levels are Public, Protected, Private, Internal and Protected Internal.

**Q5. What is Polymorphism?**

Polymorphism is nothing but assigning behavior or value in a subclass to something that was already declared in the main class. Simply, polymorphism takes more than one form.

**Q6. What is Inheritance?**

**Inheritance is a concept where one class shares the structure and behavior defined in another class. If inheritance applied on one class is called Single Inheritance, and if it depends on multiple classes, then it is called multiple Inheritance.**

**Q7. Define a constructor?**

**Constructor is a method used to initialize the state of an object, and it gets invoked at the time of object creation. Rules for constructor are:**

**• Constructor Name should be same as class name.**

**• Constructor must have no return type.**

**Q8. Define Destructor?**

**Destructor is a method which is automatically called when the object ismade ofscope or destroyed. Destructor name is also same asclass name but with the tilde symbol before the name.**

**Q9. What is function overloading?**

**Function overloading is defined as a normal function, but it has the ability to perform different tasks. It allows creation of several methods with the same name which differ from each other by type of input and output of the function.**

**Example**

**void add(int& a, int& b);**

**void add(double& a, double& b);**

**void add(struct bob& a, struct bob& b);**

**15. What is operator overloading?**

**Q10. Difference between overloading and overriding?**

Overloading is static binding whereas Overriding is dynamic binding. Overloading is nothing but the same method with different arguments , and it may or may not return the same value in the same class itself.

Overriding is the same method names with same arguments and return types associates with the class and its child class.

**Q11. Difference between class and an object?**

An object is an instance of a class. Objects hold any information , but classes don’t have any information. Definition of properties and functions can be done at class and can be used by the object.

Class can have sub-classes, and an object doesn’t have sub-objects.

**Q12. What is an abstraction?**

Abstraction is a good feature of OOPS , and it shows only the necessary details to the client of an object. Means, it shows only necessary details for an object, not the inner details of an object. Example – When you want to switch On television, it not necessary to show all the functions of TV. Whatever is required to switch on TV will be showed by using abstract class.

**Q13. What are access modifiers?**

Access modifiers determine the scope of the method or variables that can be accessed from other various objects or classes. There are 5 types of access modifiers , and they are as follows:.

* Private.
* Protected.
* Public.
* Friend.
* Protected Friend.

**Q14. How can we call the base method without creating an instance?**

Yes, it is possible to call the base method without creating an instance. And that method should be Static method. Doing inheritance from that class.-Use Base Keyword from derived class.

**Q15. What are the various types of constructors?**

There are three various types of constructors , and they are as follows:.

–  Default Constructor – With no parameters.

–  Parametric Constructor – With Parameters. Create a new instance of a class and also passing arguments simultaneously.

–  Copy Constructor – Which creates a new object as a copy of an existing object.

**Q16. What is early and late binding?**

Early binding refers to assignment of values to variables during design time whereas late binding refers to assignment of values to variables during run time.

**Q17. What is the difference between structure and a class?**

Structure default access type is public , but class access type is private. A structure is used for grouping data whereas class can be used for grouping data and methods. Structures are exclusively used for data and it doesn’t require strict validation , but classes are used to encapsulates and inherit data which requires strict validation.

**Q18. What is the default access modifier in a class?**

The default access modifier of a class is default in Java while private in C++.

**Q19. What is dynamic or run time polymorphism?**

Dynamic or Run time polymorphism is also known as method overriding in which call to an overridden function is resolved during run time, not at the compile time. It means having two or more methods with the same name,same signature but with different implementation.

**Q20. Whether static method can use non static members?**

**False.**

**Q21. What are base class, sub class and super class?**

Base class is the most generalized class , and it is said to be a root class.

Sub class is a class that inherits from one or more base classes.

Super class is the parent class from which another class inherits.

**Q22. What is static and dynamic binding?**

Binding is nothing but the association of a name with the class. Static binding is a binding in which name can be associated with the class during compilation time , and it is also called as early Binding.

Dynamic binding is a binding in which name can be associated with the class during execution time , and it is also called as Late Binding.

**Q23. How many instances can be created for an abstract class?**

Zero instances will be created for an abstract class.

**Q24. Which OOPS concept is used as reuse mechanism?**

Inheritance is the OOPS concept that can be used as reuse mechanism.

**Q25. Which OOPS concept exposes only necessary information to the calling functions?**

Encapsulation