1. What is the main difference between a class and an object?

Object: object is anything which has state and behaviour.

Eg: A dog is an object, it has state which means breed, color, age etc and behaviour or properties like eat, sleep. Run etc.

Class: A class is something which can be defined as a blueprint that describes the behaviour that the object of its type support.

1. What is Encapsulation? Explain with a used case?

Encapsulation is the type of mechanism in which the variables of a class will be hidden from other classes and can be access only through the methods of their current class. It is also known as data hiding.

If class A has 3 variables passed and then if class B wants use those variables then class B can use those only through getters and setters.

1. What is Polymorphism? Explain with a used case?

Polymorphism means having many forms. This allows perform a single action in amny different ways.

In real life a person can have many roles father, brother, employee but all the roles are done by one person. Poly means many and morph means forms.

1. Explain Overriding & Overloading and its advantages?

Overloading:

Method overloading:

In this multiple functions can have same name but different parameters then these functions are referred to as overloaded. These functions can be overloaded by changing type of arguments or number in arguments.

Operator overloading:

In this an performs differently based on the datatype it was passed with.

“+” operator is used to add when passed with integers and used to concatenate when used with string.

Overriding: It is an ability define a behaviour that’s specific to the subclass type, which is a subclass can implement apparent class method based on its requirement.

1. What is Inheritance and different types of inheritance? Explain with a used case

Inheritance is the property of an object to acquire all the properties and behaviours of its parent object.

Types of inheritance:

Single: this inherits properties and behaviours from single parent.

Multilevel: In this child class inherits properties and behaviours from parent class which is also derived from other class.

Hierarchical: In tis child inherits properties and behaviours from one or more parents.

1. What is an abstract class?

Abstract class similar to interface except that it can contain default method implementation. An abstract class can have both abstract method without body and it can have methods with implementation also.

1. What is an interface and how multiple inheritance is achieved with this?

An interface contains variables and methods like a class but the methods in an interface are abstract by default unlike a class

Multiple inheritance by interface occurs if a class implements multiple interfaces or also if an interface itself extends multiple interfaces.

1. What are the access modifiers?

The access modifiers in Java specifies the accessibility of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying the access modifier on it.

The four types of access modifiers are : Private, Default, Protected and public.

1. What are the various types of constructors?

Default constructor: In this if there is no constructor in your class, java compiler inserts a default constructor in to our class.

No-arg constructor: Constructor with no arguments is known as no-arg constructor.

Parameterized constructor: Constructor with arguments or certain parameters is known as parameterized constructor.

1. What is ‘this’ pointer?

“this” keyword is a reference variable in java .

It is used to refer the instance variable of the current class.

It can be used to initiate current class constructor.

1. What is static and dynamic Binding?

Static binding: The binding that can be resolved at the compiler level is called as static binding or early binding.

Dynamic binding: When compiler is not able to resolve the call at compiler level is known as dynamic binding.

1. How many instances can be created for an abstract class and why?

Zero.  it is not possible to create an abstract class because you do not have all of the right implementations to make it happen. The abstract class serves another purpose which is to make sure that you have a base for all of your subclasses.

1. Which OOPS concept is used as a reuse mechanism and explain with a use case?

Inheritance can be used as a reuse mechanism.

Because in inheritance there is a chance for code reusability when the child inherits from parent there is no need to write the code which is in the parent class again, instead we can just inherit which works as reusing the same code.