**JAVA Basics Interview questions:**

**What is the difference between JRE, JDK and JVM?**

JDK is an acronym of Java Development Kit. JDE is an environment that allows the developer to write programs that are readable by human, which we can understand. JDK has some development tools that facilitates a programmer achievement such as create jar archive, compile the program, modifying. In the other hand we have JRE for Java Runtime Environment. This as name applies help the programmer run his or her program to check if the program runs without errors with help of JVM. In this stage the programmer cannot add or modify the program or any application. JRE has JVM and library classes. JVM is a Java Virtual machine which comes bundle with JRE, are like hands to hand. JVM is a Java platform because it executes the program by converting the code to byte code that converts from human language to machine language.

JRE, JDK and JVM go together we cannot exclude one from another. Are like human body if one organ is missing the body will no functionate right.

**What is the Entry point in JAVA program?**

The entry point in Java is **main,** which is public static void. Without main method the JVM platform will not execute the program.

**What does static keyword mean in JAVA?**

Static keyword means in Java that a particular method belongs to itself. A method that has a static keyword does not need to instantiate an instance to be called in other static method. In the other word, whenever we see static method that means it belongs to a class rather than an instance.

**What is Object and Class in JAVA?**

Object or instance are unchangeably used. Object/ instance is used to access a non-static method what were created out of main method. We call object/ instance references type because does not belong to a particular class.

Class in java is important because is where we instantiate data types, methods, and call static and non-static methods. Each method must be inside a class to execute a well healthy program.

**What are Immutable objects in JAVA?**

Immutable objects on Java mean that after an object is created, we cannot change it. For example, Strings.

**What is this keyword in JAVA?**

The keyword “this” is a way to differentiate between an attributes and parameters. The ambiguity could create a such havocs. This keyword takes care of the ambiguity.

**What is a Constructor in JAVA?**

A constructor is method that has the same name as a class name. We use constructor to initialize primitive type in it. There are two types of constructors. Default constructor and parameterized constructor.

Default constructor, we initialize and declare primitive type and we can only have one default constructor. Parameterized constructor is where we declare our own data type that matches class’s primitives. Here where we use this keyword to differentiate between class’s data type and our data type in case, we used the same variable names. In parameterized constructor we can have as many as parameterized constructor.

**Can you inherit constructor from another class?**

Constructor cannot be inherited from anther class. Because whenever a class (child class) extends from another class (parent class), the sub class (child class) inherits all states and behaviors in the form of variables and methods from its super () class but does not inherit parent’ constructor. The reason why cannot be inherited because constructors are special and has the same name as class name. Another reason is that we cannot create an instance to call constructor because of the same reason has the same name as class name.

**What are different OOPS Concepts?**

Different Object-Oriented- Programing concepts are Class, Objects, Data Abstraction, Encapsulation, Inheritance, Polymorphism, Dynamic Binding, and Message Passing

**What are different access modifiers in JAVA?**

In java there are four different access modifiers. Default access modifier is accessible within the package and can be imported to other packages. Public access modifier is accessible within the class and outside that class. Private access modifier is only accessible to the current class that we work on. And protected access modifier is accessible to within a class and child class.

**How do you implement inheritance in JAVA? Give me an example**

We can implement inheritance in Java by using the word **extends** for example:

public childClass extends parentClass {

………

}

**How do you refer to parent class objects in JAVA?**

We refer to parent class in Java as known as super class. We used new to create a parent class object. For example: parentClass pC = new parentClass();

**What class is the superclass of all classes in JAVA?**

The supper class of all the classes in Java is a parent class. Parent class is known as super class.

**Is multiple inheritance supported in JAVA? Why is it not supported?**

Multiple inheritance is not supported by java. This term will arise a lot of problems because multiple inheritance will have multiple parent classes name ambiguity. For example, let’s say we have five parent classes (super class) that have same method name, when we will invoke a particular method using super (), will not be able to decide from which to what method to clall.

**How can you achieve multiple inheritance in JAVA?**

We can achieve multiple inheritance by implementing interface. Because One class (child class or sub class) can inherit from one class (parent class/super class) but can inherit multiple interfaces.

**How do you achieve Encapsulation?**

To achieve the encapsulation first, we must make all the attributes private using access specifier to prevent someone access these attributes outside the class. Second, create setter and getter methods inside the class to get these private variables. For example, we have

class x {

private int x1;

…….

}

public int getX(){

Return x1;

}

public void setX(int x2){ public void setX(int x1){

x1 = x2 this.x1 = x1

} }

We use this if the attribute and set method has the same variable.

**What is Polymorphism?**

Polymorphism is a magic feature that allows a programmer to write a program in many ways of coding. With out polymorphism for example whenever the program executes, we expect the same output. With polymorphism, we can create as much as many inputs that will associate with different version. overriding and overloading are part of the polymorphism.

**What is the difference between Overriding and OverLoading?**

The difference between overriding and overloading is that overloading gives the programmer the ability to create multiple methods performing multiple tasks. These multiple methods each one has the same method name and different parameters. In the other hand, we have overriding which is already existed in parent class (super class) and is used to modify to the method in child class (sub class). Overriding methods have the same exact name and parameters.

**Can we overload a constructor?**

Yes, we can overload a constructor. A constructor overloading is defined in the concept of having the same method name with different parameters, in this case each method performs a multiple task.

**Can we override static methods in JAVA?**

We cannot override static methods, because each one of these methods is called in a different stage. Overriding methods is based on dynamic binding at run time. Static methods are class level which are bonded using static binding during compile time.

**Can we override private methods in JAVA?**

We cannot override private methods in java. With the access modifier “private”, the methods are not overridden because the private methods are only accessible within the class scope which they are declared.

**What is Abstraction?**

Abstraction is one of the concepts of object-oriented-programming. Abstract class is hiding the implementation and only showing method definition. A good example of abstraction is database, we do not know how data is created, stored, and maintained. It only displays what the used need to see. This will achieve security.

**Can you make an object out of an Abstract class? How?**

We cannot create an object of an abstract class. Abstract is restricted class and cannot be used to create an object. To access an abstract class must be inherited from another class (child class/super class). What we can do is to create a reference variable of an abstract class and use that reference to refer to objects that are inherited from the super class. An abstract class that has at lest one abstract method () is using overriding.

**Can there be an abstract method without an abstract class?**

No cannot be, because in abstract method () does not perform any function. Abstract method () has an empty body. When we use abstract method (), abstract class must be declared.