**OOPS:**

**What is Abstraction?**

* Hiding internal implementation and sharing set of services is called as abstarction.
* We can achieve abstraction by using “Interface” and “Abstarct Class”.
* We can achieve 100% abstraction using “Interface” and partial by using “Abstract Class”.
* We can achieve security using abstraction.
* E.g. ATM machine, Car, Mobile, etc.

**What is Encapsulation?**

* Wrapping of data member and methods called as Encapsulation.
* We can achieve it by making data members “private”.
* POJO class is good example of encapsulation.
* In a class if it has every data member as a “private” then such class is called as tightly encapsulated.
* E.g. Engine, Gear box within Car,etc.

**Difference between Abstraction and Encapsulation?**

|  |  |
| --- | --- |
| Abstraction | Encapsulation |
| Hiding internal implementation and sharing  set of services. | Wrapping of data member and methods. |
| We can achieve abstraction by using “Interface” and “Abstarct Class”. | We can achieve it by making data members “private”. |
| It increases code. | It decreases code. |
| It solves problem at Design level. | It solves problem at implementation level. |

**What is Inheritance?**

* Acquiring properties of Parent class is called inheritance.
* It is also called as IS-A relationship.
* It can be achieved by using “extends” keyword, by making Parent-Child relationship.
* It helps in reusability of code.
* E.g. New Car version inherits properties from old versions.

**What is Polymorphism?**

* It means one name many forms.
* It makes reusability simple and also makes code understanding easy.
* There are two types of Polymorphism :
* Run-time Polymorphism (Dynamic Binding, Overriding)
* Decision making at Runtime by using runtime object.
* Used for adding additional functionality into existing one.
* Useful only in Parent-Child Relationship.
* Compile-time Polymorphism (Static Binding, Overloading)
* Best example of polymorphism is “println” method of “printstream” class.

**Difference between IS-A relationship & HAS-A relationship?**

|  |  |
| --- | --- |
| IS-A relationship | HAS-A relationship |
| It is also known as inheritance, it is  acquiring parent class properties. | It is acquiring class properties by creating  instance of that class. |
| It is achieved by “extends” keyword. | It is achieved by “new” keyword. |
| Method with same name, signature but different return types can’t be created in  parent-child class. | Method with same name and different return type can be created. |

**Difference between Overloading & Overriding?**

|  |  |
| --- | --- |
| Overloading | Overriding |
| Decision making done at compile-time, where compiling is based on reference  type. | Decision making done at run-time, where JVM run is based on runtime object. |
| We can overload method and constructor  both. | We can override method but we can’t  override constructor. |
| We can overload static, private and final  method. | We can’t override static, private and final  method. |
| E.g. “println” method of Printstream  class. | E.g. ‘equals’ method of object class and  ‘equals’ method of String class. |

**What is Static?**

* Static is keyword.
* We can apply static keyword with variables, methods, blocks and classes.
* The static variable gets memory only once in the class area at the time of class loading. It can be used to refer to the common properties of all objects, for example, the company name of employees, college name of students, etc.
* A static method belongs to the class rather than the object of a class. It can be invoked without the need for creating an instance of a class. It can access static data member and can change the value of it. It can be used to setup database connection.
* Static block is used to initialize the static data member. It is executed before the main method at the time of class loading.

**What is non-static block?**

* It is used for non-initializing content.
* Before calling constructor non-static block is executed.

**What is constructor?**

* A constructor in Java is a special method that is called when an object of a class is created.
* It is used to initialise object as well as non-static variables.
* Constructors can also take parameters to initialize data members.

**What is Object?**

* An object is an instance of a class.
* Objects have states and behaviours. Example: A dog has states - color, name, breed as well as behaviours – eat, bark, smell.
* Ways we can create an object :
* By using new Operator : Test t = new Test();
* By using newInstance() :(Reflection Mechanism) Test t=(Test)Class.forName("Test").newInstance();
* By using Clone() : Test t1 = new Test();
* Test t2 = (Test)t1.clone();
* By using Factory methods : Runtime r = Runtime.getRuntime();
* DateFormat df = DateFormat.getInstance();
* By using Deserialization :
* FileInputStream fis = new FileInputStream("abc.ser");