**Object Creation:**

* Using new keyword (Student s1 = new Student() )
* Using new instance method

Student s1 = (Student) class.forName(“"JavaIntQA.Student"”).newInstance();

Student s2 = Student.class.newInstance();

There is a class called constructor

* Class class newInstance() method can only invoke no argument constructor or default constructor
* Constructor class newInstance() method can invoke any constructor
* Class class newInstance() method can only invoke visible constructors
* Constructor class newInstance() method can invoke private constructors also in come classes
* Class class newInstance() method throws all the exceptions that are thrown by the constructor
* Constructor class new Instance() method wraps the thrown exceptions with an InvocationTarget exception
* Spring,Hibernate frameworks internally call **constructor class new instance** method to create objects.This way of object creation is called reflective way of object creation in java

**Constructor<Student> cons = Student.class.getConstructor();**

**Student s2 = cons.newInstance();**

* While creating an object using clone() method ,**constructor** is **not called**
* Before trying to create an object using clone method, the class to which u want to create an object, should implement the **Cloneable** interface and override the clone method
* Student s3 = (Student)s2.clone();
* The last way to create object is Deserialization.In this way of object creation also,constructor is not getting called.
* The class for which object needs to be created should implement the Serializable interface

**FileInputStream fis = new FileInputStream("Text.txt");**

**ObjectInputStream Ois = new ObjectInputStream(fis);**

**Student s5 = (Student)Ois.readObject();**