Object in Java

Description

OBJECT is an instance of a class. An object is nothing but a self-contained component which consists of methods and properties to make a particular type of data useful. When you send a message to an object, you are asking the object to invoke or execute one of its methods as defined in the class.

From a programming point of view, an object can include a data structure, a variable, or a function. It has a memory location allocated. The object is designed as class hierarchies.

An object consists of:

1. **State**: It is represented by attributes of an object. It also reflects the properties of an object.
2. **Behaviour**: It is represented by methods of an object. It also reflects the response of an object with other objects.
3. **Identity**: It gives a unique name to an object and enables one object to interact with other objects.

Ways to create object of a class

1. Using new keyword:

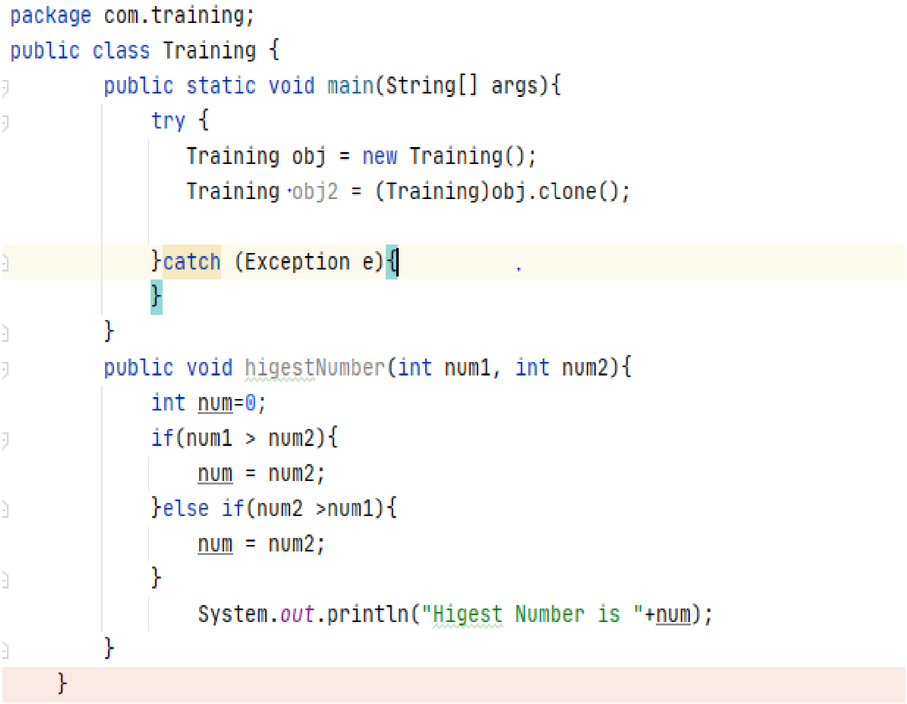
It is the most common and general way to create object in java. Example:

```
// creating object of class Test

Test t = new Test();
```

2. Using clone () method:

clone() method is present in Object class. It creates and returns a copy of the object.



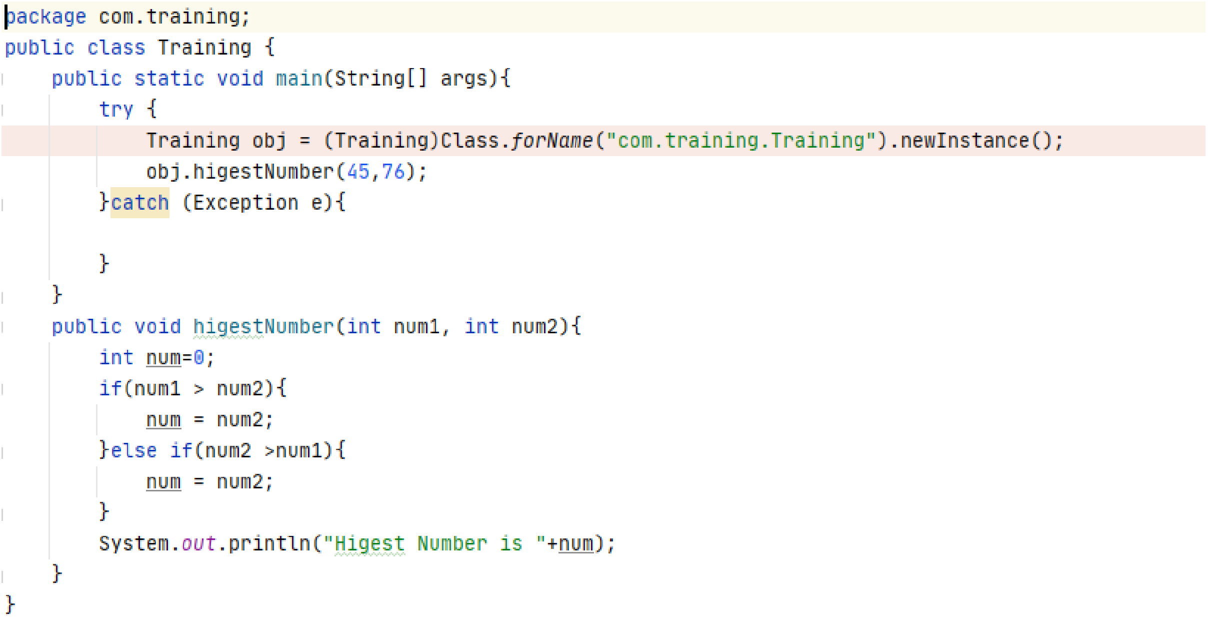
```
package com.training;
public class Training {
        public static void main(String[] args){
            try {
               Training obj = new Training();
               Training obj2 = (Training)obj.clone();

            }catch (Exception e){
            }
        }
        public void higestNumber(int num1, int num2){
            int num=0;
            if(num1 > num2){
                num = num2;
            }else if(num2 >num1){
                num = num2;
            }
                System.out.println("Higest Number is "+num);
        }
    }
```

3. Using Class.forName(String className) method :

There is a pre-defined class in java.lang package with name Class. The forName(String className) method returns the Class object associated with the class with the given string name.We have to give the fully qualified name for a class. On calling new Instance() method on this Class object returns new instance of the class with the given string name.

Program:



```java
package com.training;
public class Training {
    public static void main(String[] args){
        try {
            Training obj = (Training)Class.forName("com.training.Training").newInstance();
            obj.higestNumber(45,76);
        }catch (Exception e){

        }
    }
    public void higestNumber(int num1, int num2){
        int num=0;
        if(num1 > num2){
            num = num2;
        }else if(num2 >num1){
            num = num2;
        }
        System.out.println("Higest Number is "+num);
    }
}
```

Note:

A **class** is a **blueprint** or **prototype** that defines the variables and the methods (functions) common to all objects of a certain kind.

An Object is a specimen of a class. Software objects are often used to model real-world objects you find in everyday life.