Website
https://www.curiousinfotechsolutions.com



# Java Programming

## Day-8


| Class & Object | https://youtu.be/uN1SIx33C7c |
|---|---|

Email id:
**info@curiousinfotechsolutions.com**

# Syllabus Day-8




Types of Java Variables

Create an object

Create Multiple Objects

Using Multiple Classes in one file

Using private Data Member

Using Multiple Classes file

Class and Object: Read & Write Student Detail

# Types of Java Variables




- 1) Local Variable
  - A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.
  - A local variable cannot be defined with "static" keyword.
- 2) Instance Variable
  - A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as static.
  - It is called an instance variable because its value is instance-specific and is not shared among instances.
- 3) Static variable
  - A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class. Memory allocation for static variables happens only once when the class is loaded in the memory.

# Types of Java Variables



```
public class Main
{
   static int m=100;//static variable
   static void method()
   {
      int n=90;//local variable
      m = m + 10;//110
      System.out.println(n + m); //2->200
   }
   public static void main(String args[])
   {
      int data=50;//instance variable
      System.out.println("Data = "+ data);//1->Data=50
      method();
      System.out.println("m="+m);//3->m=110
   }
}//end of clas
```

There are three types of Java Variables

- local variable
- instance variable
- static variable

```
1.public class Main
2.{
3.   static int m=100;//static variable
4.   void method()
5.   {
6.      int n=90;//local variable
7.   }
8.   public static void main(String args[])
9.   {
10.      int data=50;//instance variable
11.
12.   }
13.}//end of class
```

# Create an object called "myObj" and print the value of x:



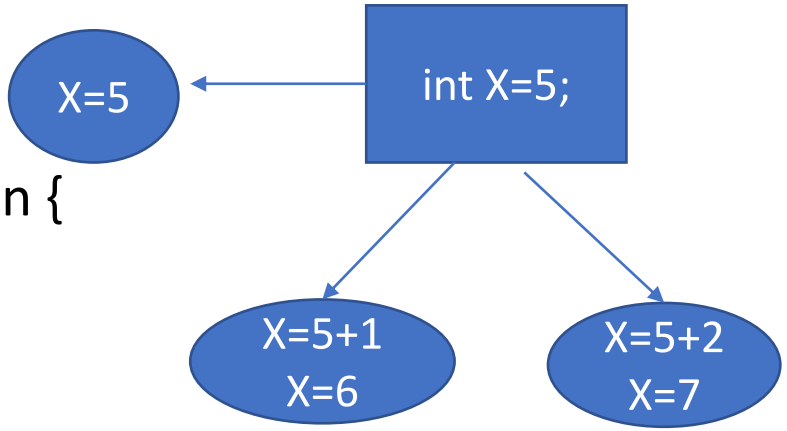

```
public class Main {
  int x = 5;


  public static void main(String[] args) {
    Main myObj = new Main();
    System.out.println(myObj.x);
  }
}
```

In Java, an object is created from a class. We have already created the class named `Main`, so now we can use this to create objects.

To create an object of `Main`, specify the class name, followed by the object name, and use the keyword `new`:

# Multiple Objects






```
pub class Main {
  int x = 5;

  public static void main(String[] args) {
    Main myObj1 = new Main();  // Object 1
    Main myObj2 = new Main();  // Object 2
    System.out.println(myObj1.x);//x=5,x=6
    System.out.println(myObj2.x);//x=5,x=7
  }
}
```

# Using Multiple Classes in a program



```
class Second {
  int x = 5;
}

class Main {
  public static void main(String[] args) {
   Second myObj = new Second();
    System.out.println(myObj.x);
  }
}
```

# Using Multiple Classes in two file



```
class Second {
  int x = 5;
  void writeData(){
   System.out.println (x);
   }
}
```

```
class Main {
  public static void main(String[] args) {
   Second myObj = new Second();
    myObj.writeData();
    System.out.println(myObj.x);
  }
}
```

# Using Multiple Classes using private DM


```
class Second {
   private int x = 5;
   void writeData(){
    System.out.println (x);
    }
}
```

```
class Main {
  public static void main(String[] args) {
   Second myObj = new Second();
    myObj.writeData();
    System.out.println(myObj.x); //error
  }
}
```

# Example of Class and Object




```
class Rectangle
{
    int length,width;
    void getData() {length = 5; width = 10;}
    void rectArea() {int area = length*width; System.out.println(area);}
}
```

```
class Main
{
    public static void main (String[] args) {
        Rectangle rectObj1 = new Rectangle();
        rectObj1.getData();
        rectObj1.rectArea();//50
    }
}
```

```
class Main
{
    public static void main (String[] args) {
        Rectangle rectObj1 = new Rectangle();
        rectObj1.length=50; rectObj1.width=100;
        int area = rectObj1.length * rectObj1.width;
        System.out.println(area);//5000
    }
}
```

# Using Multiple Classes



Main.java

```
public class Main {
  int x = 5;
}
```

Second.java

```
class Second {
  public static void main(String[] args) {
    Main myObj = new Main();
    System.out.println(myObj.x);
  }
}
```

5

# Example – Class and Object
# Read & Write Stud Detail



```
class Student{
 int id;
 String name;
 void readData(){
    id = 12345;
    name = "Vivek";
 }
 void writeData(){
    System.out.println("Stud Id:" + id);
    System.out.println("Stud Name:"+name);
 }
}
```

```
class Main{
 public static void main(String args[]){
  Student s1=new Student();
  s1.readData();
  s1.writeData();
 }
}
```