Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

SAIRAM
DIGITAL RESOURCES

YEAR III

SEM V

CS8392

OBJECT ORIENTED PROGRAMMING
(Common to CSE, EEE, EIE, ICE, IT)

UNIT NO 1

INTRODUCTION TO OOP AND JAVA FUNDAMENTALS

1.4 Defining Classes in Java

COMPUTER SCIENCE & ENGINEERING

First Program of Java

Java Programming without public

```
class cse
{
        public static void main(String args[])
        {
        System.out.println("Hello world");
        }
}
```

output

```
Hello world
```

File name: hello.java
compilation: javac hello.java
Run time : java cse

In java, both file name and class name need not be same

Java Programming with public

```
public class cse
{
        public static void main(String args[])
        {
        System.out.println("Hello world");
        }
}
```

output

```
Hello world
```

File name: cse.java
compilation: javac cse.java
Run time : java cse

When you have declared your class is public, both file name and class name should be same, any one can access this file and class publically.

Main Method

Main() method

Since every java program requires a main() as its starting point, this method is the essential part of a java program.

The main method should be present inside this class and this class name is the program file name.

Example:

```
class hello
{
public static void main(String args[])
         {
         //main body
         }
}
```

Class

```
class classname
{
type instatnce-variable1;
type instatnce-variable2;
….
type instatnce-variableN;
      type method1(parameter list)
      {
      //body
      }
      type method2(parameter list)
      {
      //body
      }
      …..   .
      type methodN(parameter list)
      {
      //body
      }
}
```

```
class cse
{
        int a,b;          // instance variables

        void display()     // method
        {
        System.out.println(“hello world”);
        }
};
```

Class and Object

Example for class:

//Program: RectArea.java

```
class Rectangle
{
//fields and method declaration;
}
class RectArea
{
public static void main(String args[])
{
Rectangle rect1=new Rectangle();  //object rect1 created
Rectangle rect2=new Rectangle(); //object rect2 created
}
}
```

Creating an Object:

There are three steps when creating an object from a class:

Declaration:

A variable declaration with a variable name with an object type.

Instantiation:

The 'new' key word is used to create the object.

Initialization:

The 'new' keyword is followed by a call to a constructor. This call initializes the new object.

Syntax for creating object

Class_name object_name = new class_name();

Example for creating object

Rectangle rect1=new Rectangle(); //object rect1 created

Class, Object and Methods

```
  //RectArea.java
class Rectangle
{
int length, width;
void getData(int x,int y)
{
   length=x;
   width=y;
}
int rectArea()
{
   int area=length*width;
   return(area);
}
}
```

```
class RectArea
{
public static void main(String args[])
 {
int a1;
Rectangle r1=new Rectangle();
r1.length=15;
r1.width=10;
a1=r1.length*r1.width;
System.out.println("Area1 " + a1);
Rectangel r2=new Rectangle();
r2.getData(10,20);
System.out.println(“Area2”+r2.rectArea());
}
}
```

Accessing Instance Variables and Methods:

Instance variables and methods are accessed via created objects.

To access an instance variable the fully qualified path should be as follows:

/* First create an object */

Syntax:

ObjectReference = new Constructor();

Example:

Rectangle r1=new Rectangle();

/* Now call a variable as follows */

Syntax:

ObjectReference.variableName;

Example:

r1.length=15;

/* Now you can call a class method as follows */

Syntax:

ObjectReference.MethodName();

Example:

r2.getData(10,20);

Video Link

https://www.youtube.com/watch?v=G1Iln3PSrUg