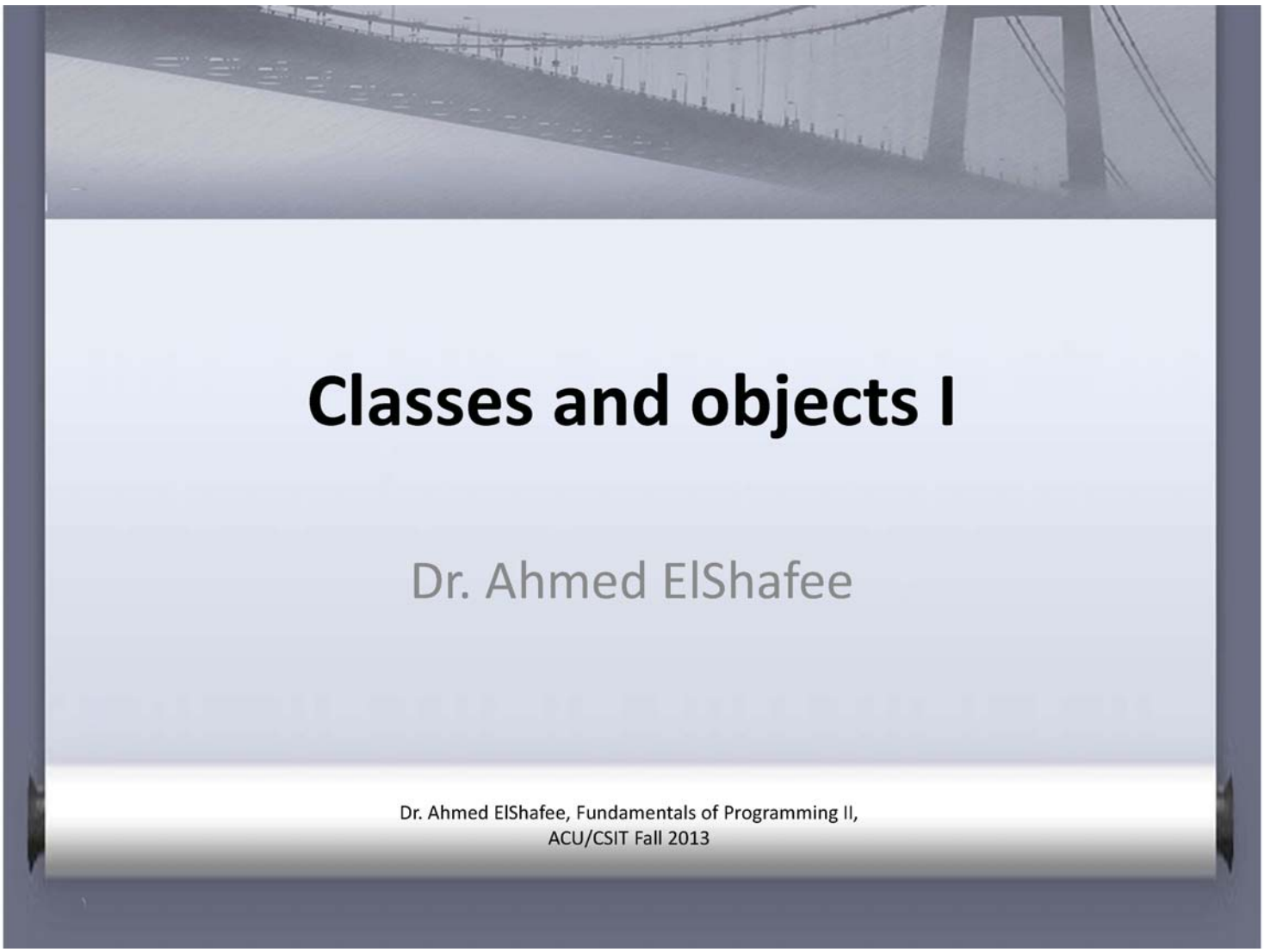


Classes and objects I

Dr. Ahmed ElShafee

Creating a Class

- A class is a technique of using one or a group of variables to be used as a foundation for a more detailed variable.
- To create a class, you start with the **class** keyword followed by a name and its body delimited by curly brackets.

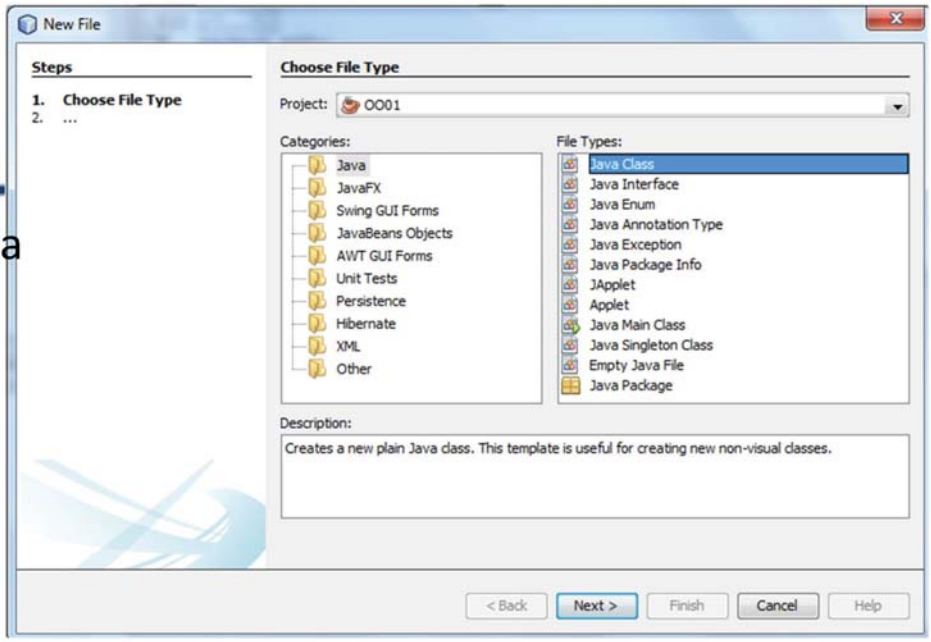
```
class PointXY
{
}
```

- A class is created in a code file.

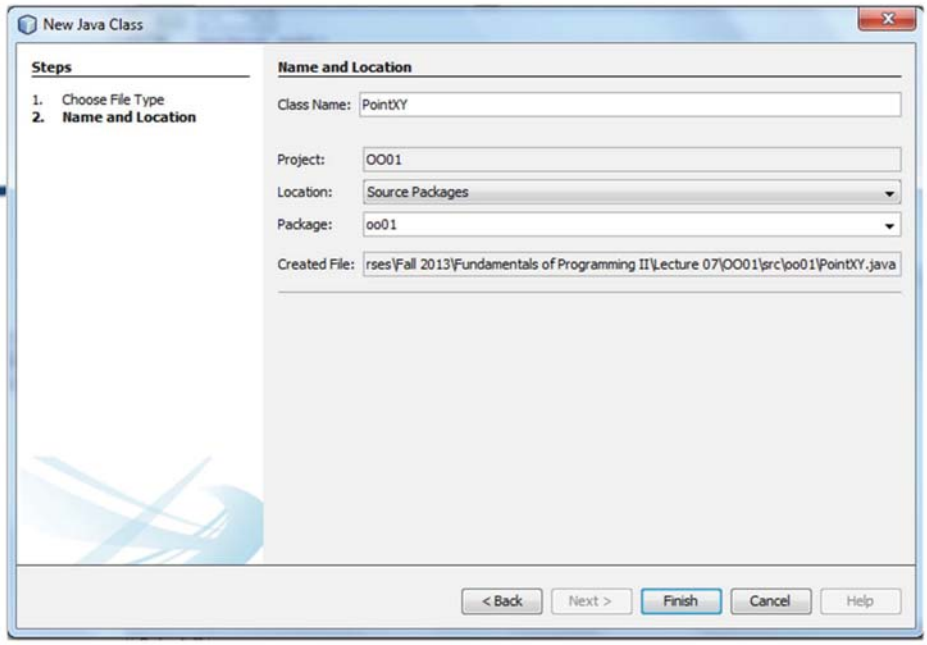
As such, you can include it in the first file of your project.

```
1  /*...*/
5  package oo01;
6
7  /**...*/
11 public class OO01 {
12
13     /**...*/
16     public static void main(String[] args) {
17         // TODO code application logic here
18     }
19 }
20
21 class PointXY
22 {
23
24 }
```

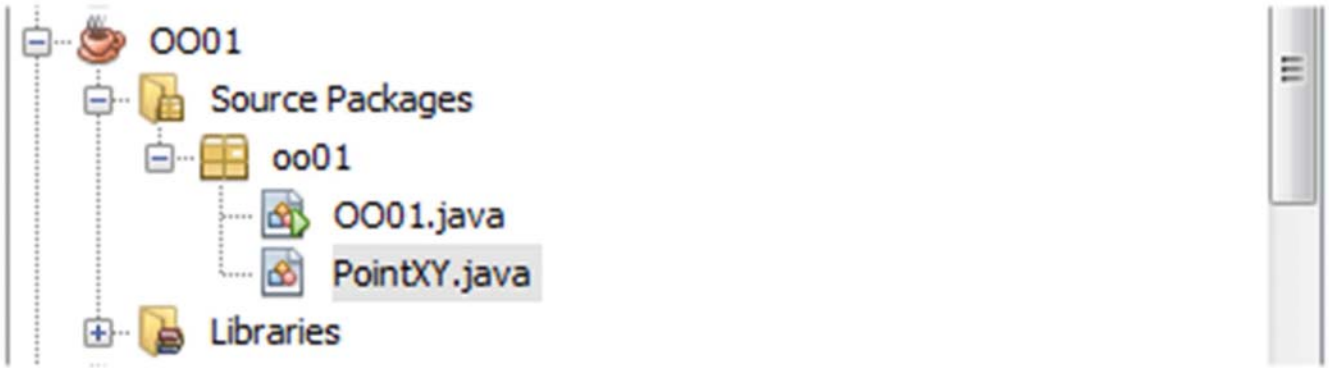
- You can also create a class in its own file.
- On the main menu, click File -> New File...
- On the File toolbar, click the New File button
- Type class name “PointXY”



- A new file contains “PointXY” class in created




```
1  /*...*/
5  package oo01;
6
7  /**...*/
11 public class PointXY {
12
13 }
14
```



Declaring a Variable of a Class Type

- Like any normal variable, to use a class in your program, you can first declare a variable for it like the variables we introduced before.
- This is not a value variable, it's an object variable.
- So you need another line, to allocate memory for it using the **new** operator.

```
1  /*...*/
5  package oo01;
6
7  /**...*/
11 public class OO01 {
12
13     /**...*/
16     public static void main(String[] args) {
17         PointXY p1;
18     }
19 }
20
21
```

```
1  /*...*/
5  package oo01;
6
7  /**...*/
11 public class OO01 {
12
13     /**...*/
16     public static void main(String[] args) {
17         PointXY p1;
18         p1=new PointXY();
19     }
20 }
21
22
```

- You can do the both steps in a single line

```
1  /*...*/
5  package oo01;
6
7  /**...*/
11 public class OO01 {
12
13     /**...*/
16     public static void main(String[] args) {
17         PointXY p1;
18         p1=new PointXY();
19         PointXY p2=new PointXY();
20     }
21 }
22
23
```


```
1  /*...*/
5  package oo01;
6
7  /**...*/
11 public class OO01 {
12
13     /**...*/
16     public static void main(String[] args) {
17         PointXY p1;
18         p1=new PointXY();
19         PointXY p2=new PointXY();
20     }
21 }
22
23
```


Sharing a Class

- When creating a class, if you want it to be accessible by code in other files (packages), precede the **class** keyword with **public** when creating it.
- If the **class** keyword is preceded by **public**, the class must be created in its own file.

```
1  /*...*/
5  package oo01;
6
7  /**...*/
11 public class PointXY {
12
13 }
14
```

Class' Member Variables

- The section between the curly brackets, { and }, of a class is referred to as its body.
- In the body of a class, you can create a list of the parts that make up the class.
- Each of these parts must be a complete variable with a name and a data type.
- For example, here are the characteristics that make up a house,

```
1  /*...*/
5  package oo01;
6
7  /**...*/
11 public class PointXY {
12     int X;
13     int Y;
14 }
15
```


Initializing an Object

- After declaring an instance of a class, you can access each of its members and assign it the desired value.

```
1  /*...*/
5  package oo01;
6
7  /**...*/
11 public class OO01 {
12
13     /**...*/
16     public static void main(String[] args) {
17         PointXY p1;
18         p1=new PointXY();
19         PointXY p2=new PointXY();
20         p1.X=1;
21         p1.Y=0;
22     }
23 }
24
25
```

- Once a member variable has been initialized, you can use the dot operator to access it and retrieve its value:

```
1  /*...*/
5  package oo01;
6
7  /**...*/
11 public class OO01 {
12
13     /**...*/
16     public static void main(String[] args) {
17         PointXY p1;
18         p1=new PointXY();
19         PointXY p2=new PointXY();
20         p1.X=1;
21         p1.Y=0;
22         System.out.println("Point 1:");
23         System.out.println("X="+p1.X);
24         System.out.println("Y="+p1.Y);
25         System.out.println(p1);
26     }
27 }
```

```
Output - OO01 (run)
run:
Point 1:
X=1
Y=0
oo01.PointXY@1e4ad14c
BUILD SUCCESSFUL (total time: 6 seconds)
```


OO01

check Lab manual

OO02

check Lab manual

The Methods of a Class

- A method is simply a section of code that takes care of a particular detail for the functionality of the class.
- To create a method, you specify its name, which follows the rules we defined for **variables**.

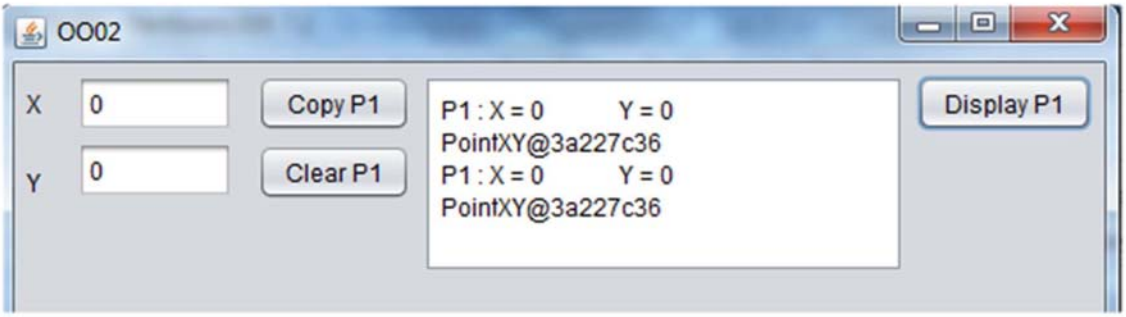
```
1  /*...*/
5
6
7  /**...*/
11 public class PointXY {
12     int X;
13     int Y;
14     public void initPointXY()
15     {
16         X=0;
17         Y=0;
18     }
19 }
20
```

- After creating a method, in its body delimited by its curly brackets, you can define the desired behavior.

```
1  /*...*/
5
6
7  /**...*/
11 public class PointXY {
12     int X;
13     int Y;
14     public void initPointXY()
15     {
16         X=0;
17         Y=0;
18     }
19 }
20
```


OO03

Check Lab manual

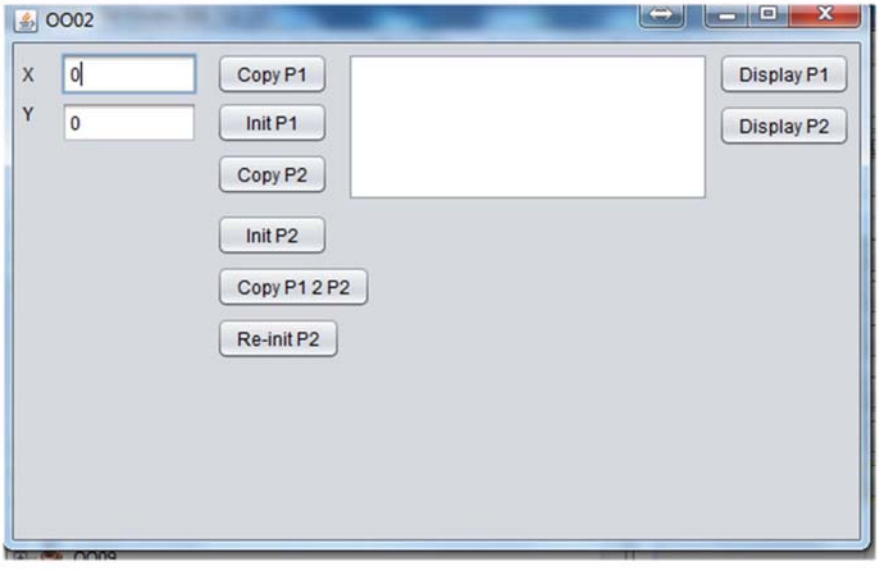


- A variable you have declared of a class is also called an instance of the class.
- In the same way, you can declare various instances of the same class as necessary:
- Each one of these instances gives you access to the members of the class but each instance holds the particular values of the members of its instance.

```
PointXY p1=new PointXY();
  PointXY p2=new PointXY();
```


OO04

Check Lab manual



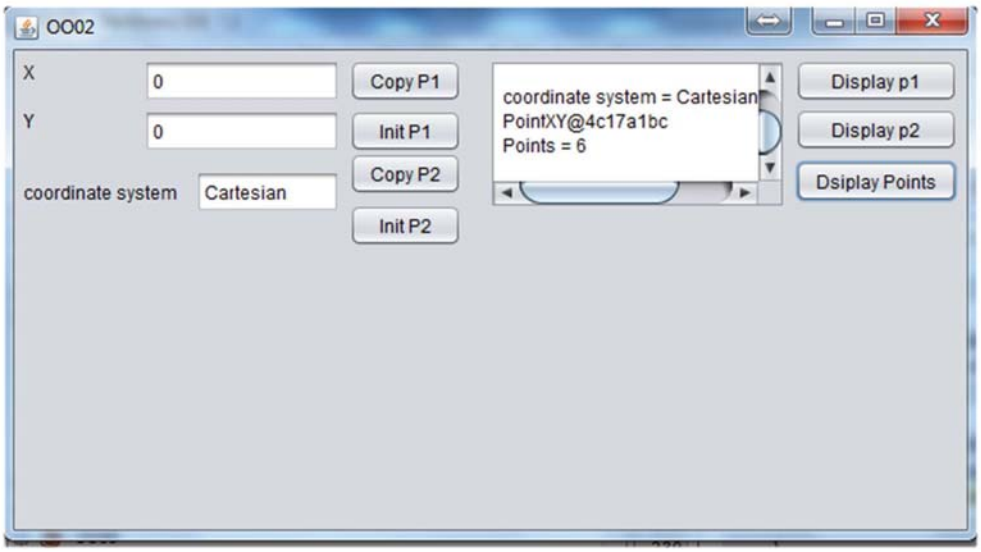
The Static Members of a Class "Static Fields"

- In your application, you can declare a class member and refer to it regardless of which instance of an object you are using. Such a member variable is called static.
- To declare a member variable of a class as static, type the **static** keyword on its left.
- Whenever you have a static member, in order to refer to it, you must "qualify" it in the class in which you want to call it.
- Qualifying a member means you must specify its class.

```
1  /*...*/
5
6
7  /**...*/
11 public class PointXY {
12     static String coordinateType;
13     int X;
14     int Y;
15     public void initPointXY()
16     {
17         X=0;
18         Y=0;
19     }
20 }
21
```


OO05

Check Lab manual



The Static Members of a Class Static Methods

- Like a member variable, a method of a class can be defined as static.
- Consequently, this particular method can access any member of the class regardless of the instance if there are many instances of the class declared.

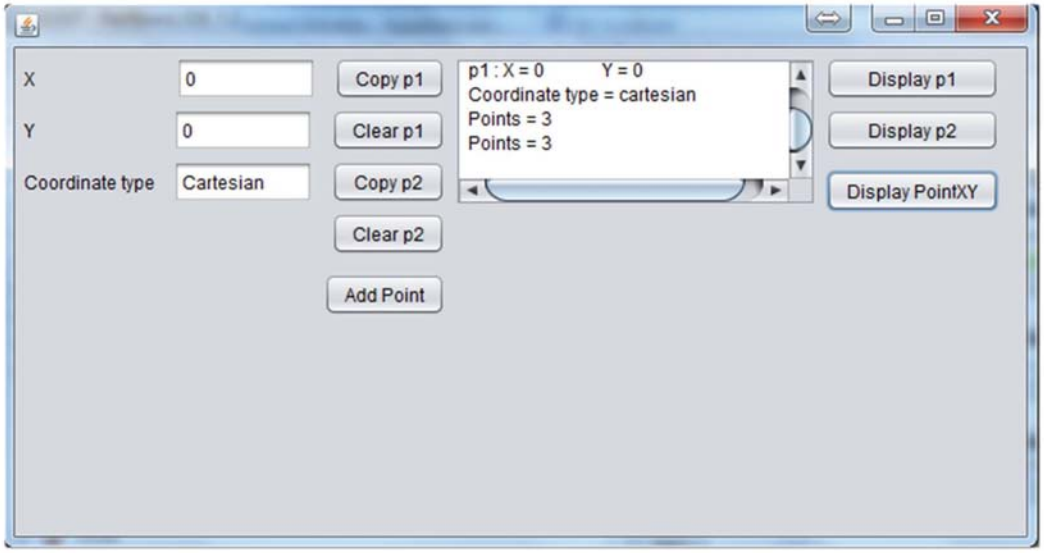
```
1  /*...*/
5
6
7  /**...*/
11 public class PointXY {
12     static String coordinateType;
13     static int X;
14     static int Y;
15     public static void initPointXY()
16     {
17         X=0;
18         Y=0;
19     }
20 }
21
```


Characteristics of Members of a Class Constants

- You can create a constant variable in a class. To create a constant variable, type the **final** keyword to the left of the variable.
- Once again, when declaring a constant, you must initialize it with an appropriate constant value.

```
1  /*...*/
5
6
7  /**...*/
11 public class PointXY {
12     final String coordinateType="cartesian ";
13      int X;
14      int Y;
15     public  void initPointXY()
16     {
17         X=0;
18         Y=0;
19     }
20 }
21
```

Check lab manual



Characteristics of Members of a Class this Instance

- If a class contains member variables and methods, the (non-static) member variables are automatically available to the method(s) of the class, even member variables that are private.
- When accessing a member variable or a method from another method of the class, to indicate that the member you are accessing belongs to the same class, you can precede it with the **this** member and the period operator.
- When using the **this** member variable (in C/C++, it is a pointer), you can access any member of a class within any method of the same class.

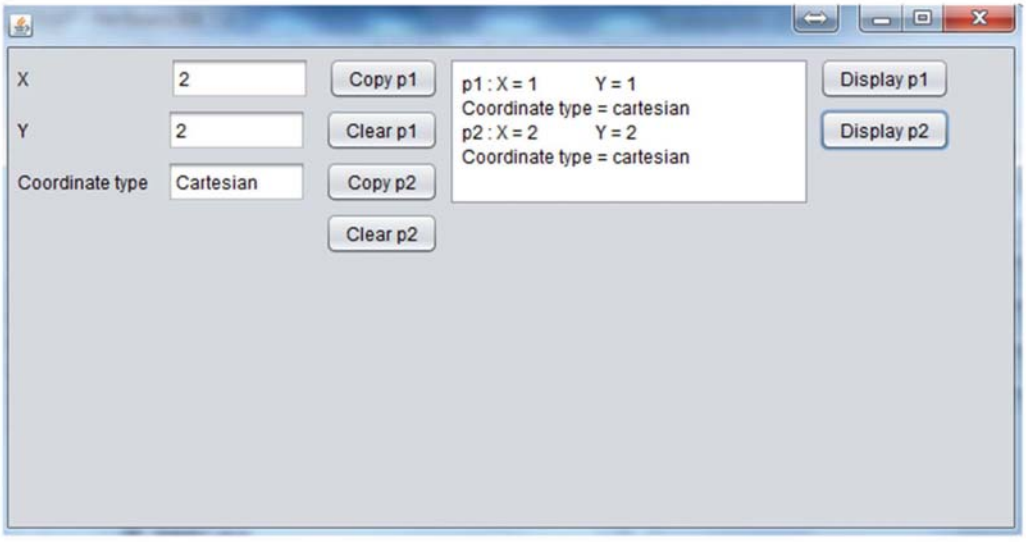
There are rules you must observe when using **this**:

- The **this** member can never be declared: it is automatically implied when you create a Class **this** cannot be used in a class A to access a member of class B.
- **this** cannot be used in a **static** method

```
1  /*...*/
5
6
7  /**...*/
11 public class PointXY {
12     public String coordinateType="cartesian ";
13      int X;
14      int Y;
15     public  void initPointXY()
16     {
17         this.X=0;
18         this.Y=0;
19     }
20 }
21
```

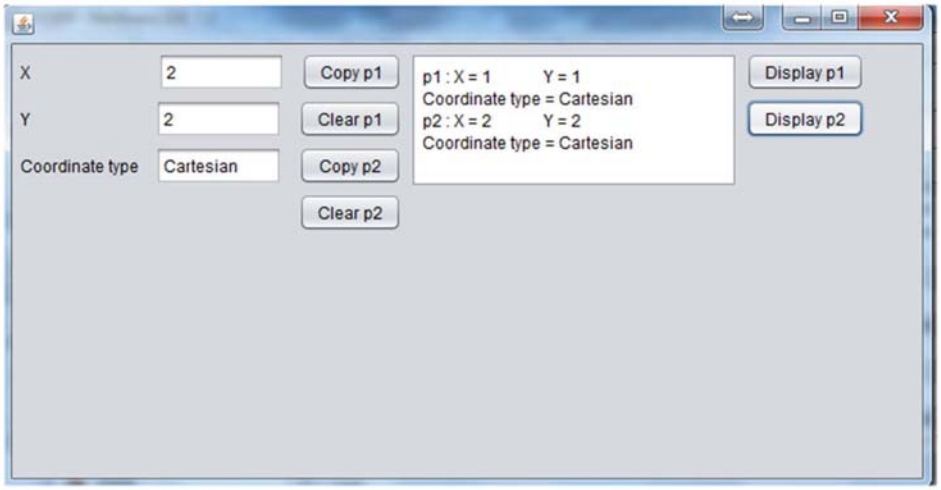

OO08

Check lab manual



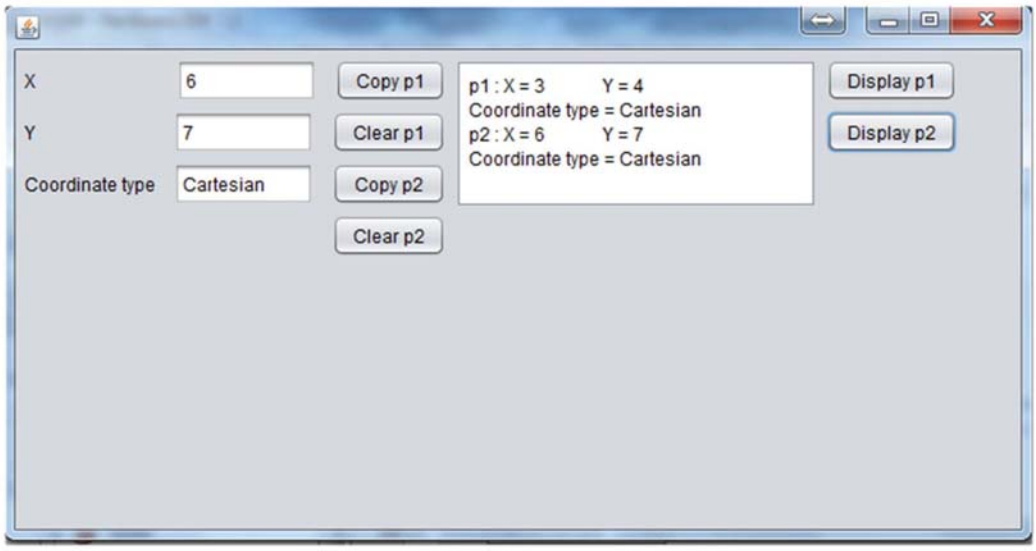
OO09 (Class return methods)

Check lab manual



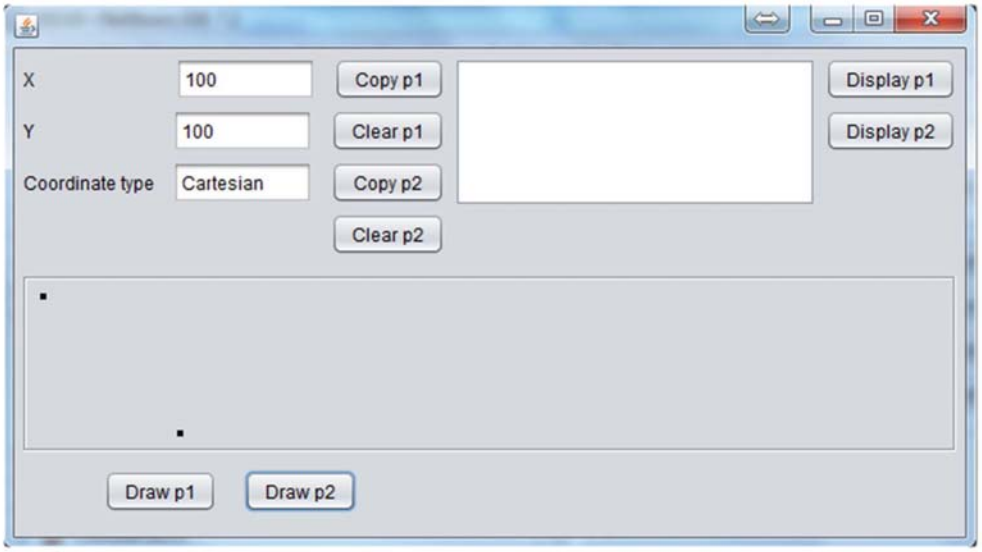
OO10 (Methods' Arguments)

Check lab manual



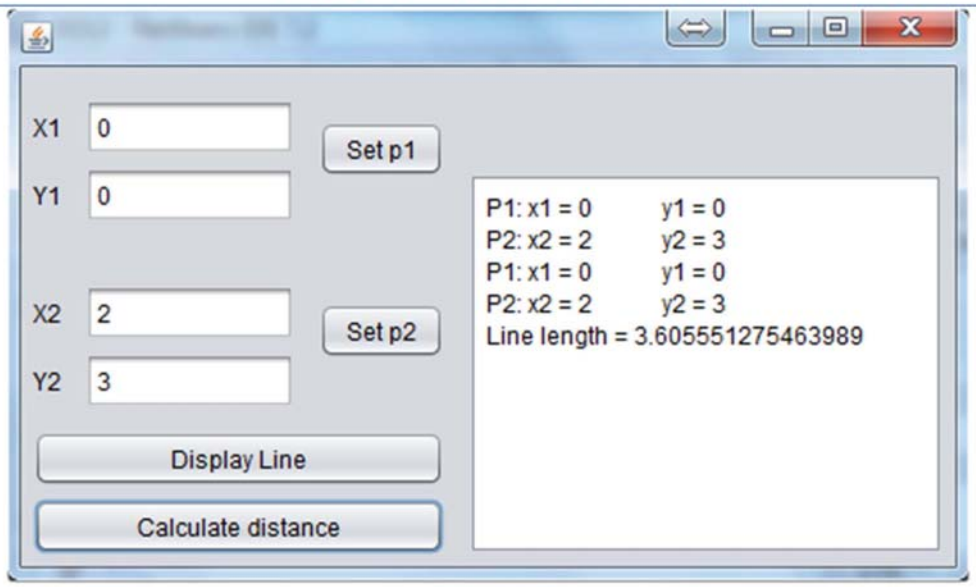
OO11

- **Check lab manual**

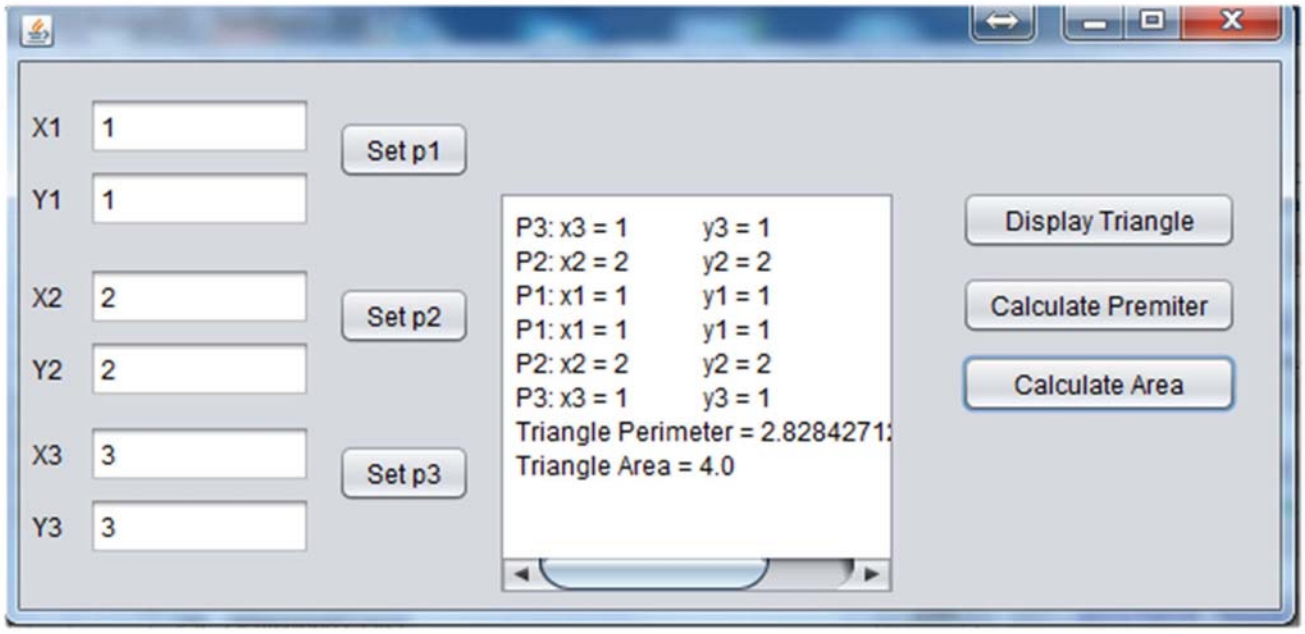


OO12

- **Check lab manual**



Assignments 1



Assignment 2

•

Thanks,

See you next Lecture, isA