

# Classes and objects I

Dr. Ahmed ElShafee

Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013

## Creating a Class

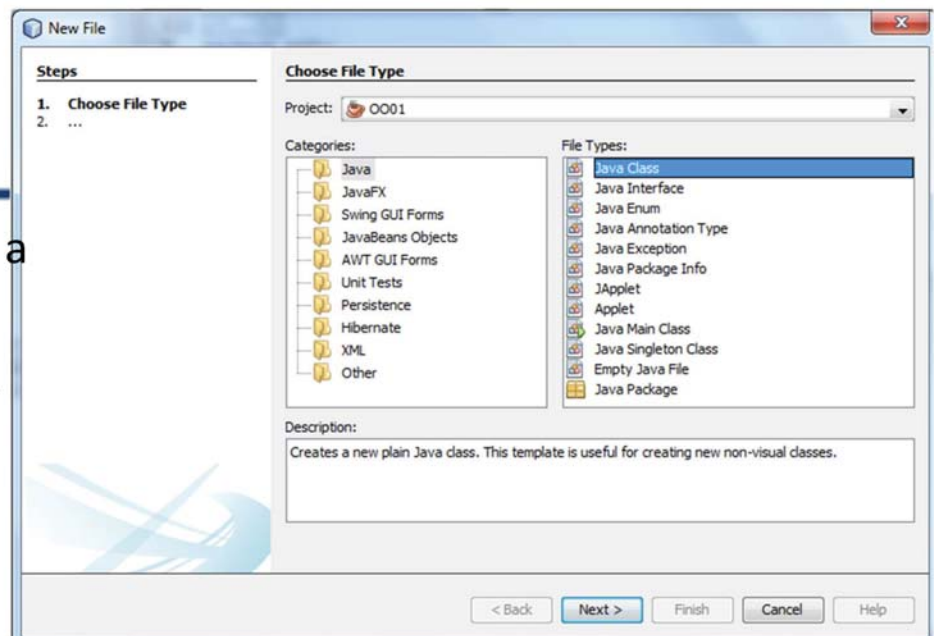
- A class is a technique of using one or a group of variables to be used as a foundation for a more detailed variable.
- To create a class, you start with the **class** keyword followed by a name and its body delimited by curly brackets.

```
class PointXY  
{  
}
```

- A class is created in a code file. As such, you can include it in the first file of your project.

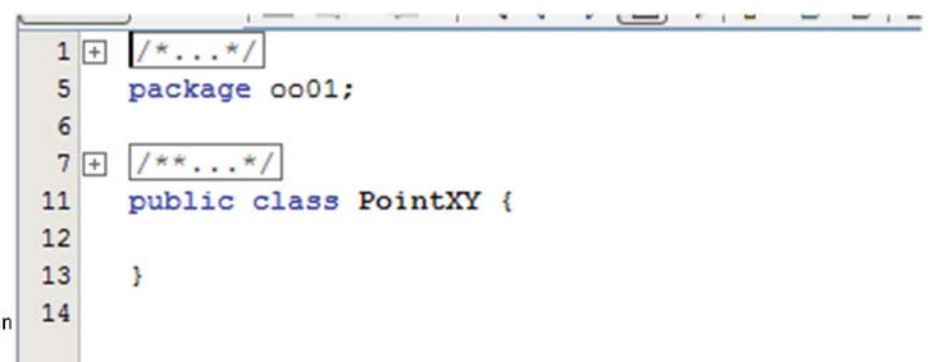
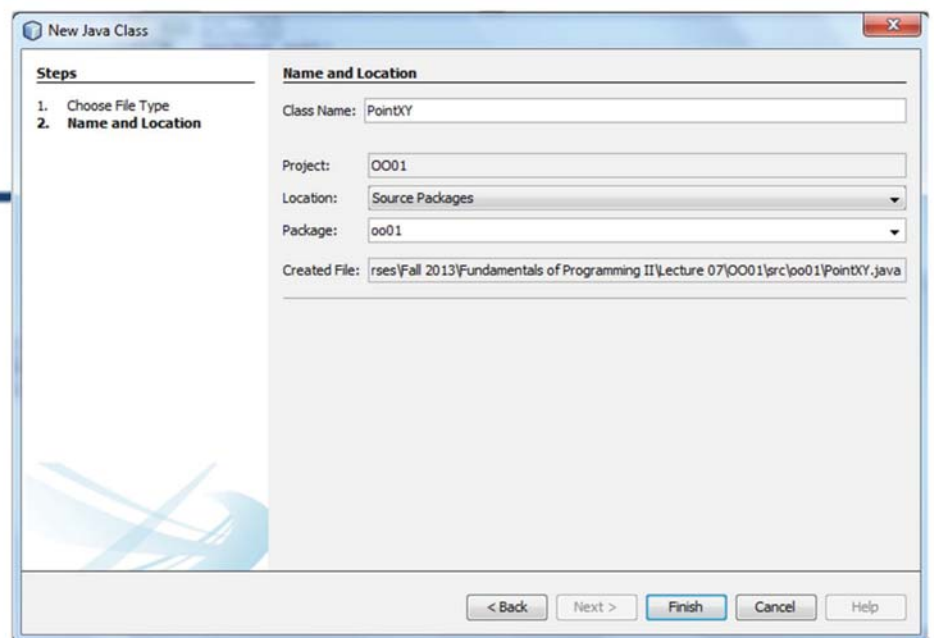
```
1  /*...*/  
5  package oo01;  
6  
7  /*...*/  
11 public class OO01 {  
12  
13     /*...*/  
16     public static void main(String[] args) {  
17         // TODO code application logic here  
18     }  
19 }  
20  
21 class PointXY  
22 {  
23  
24 }
```

- 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"

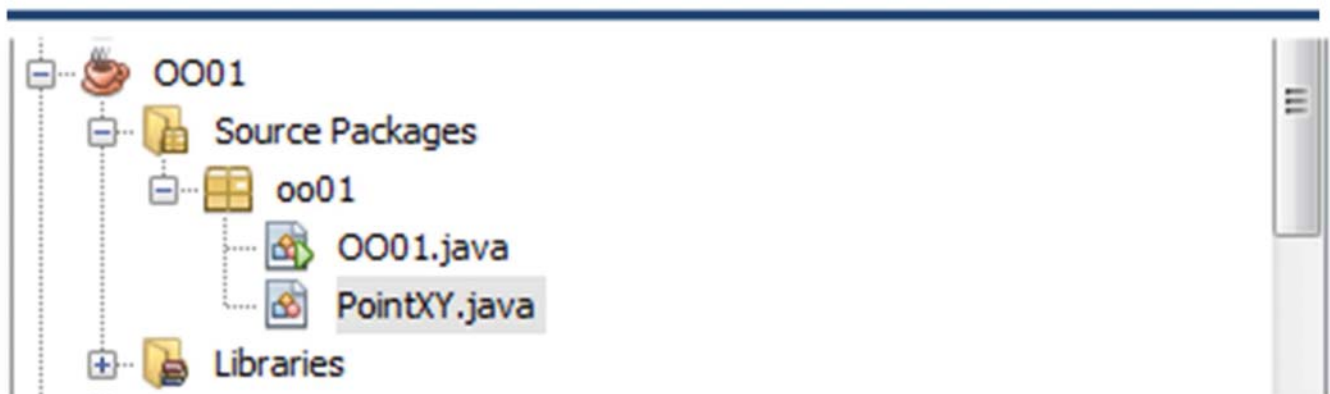


Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013

- A new file contains "PointXY" class in created



Dr. Ahn



## 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 |

```

Y

Dr. Ahmed ElShafee, Fu  
ACU/C

## 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

```

A

Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013

# 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
```

Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013

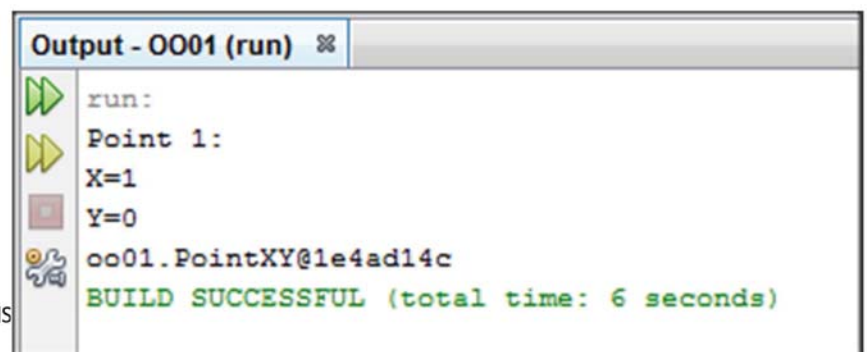
## Initializing an Object

- 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 }
```



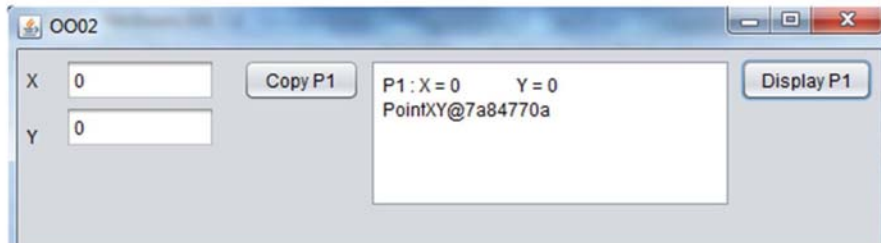
```
Output - OO01 (run) %
run:
Point 1:
X=1
Y=0
oo01.PointXY@1e4ad14c
BUILD SUCCESSFUL (total time: 6 seconds)
```

Dr. Ahmed EIS

# OO01

## check Lab manual

## 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**.

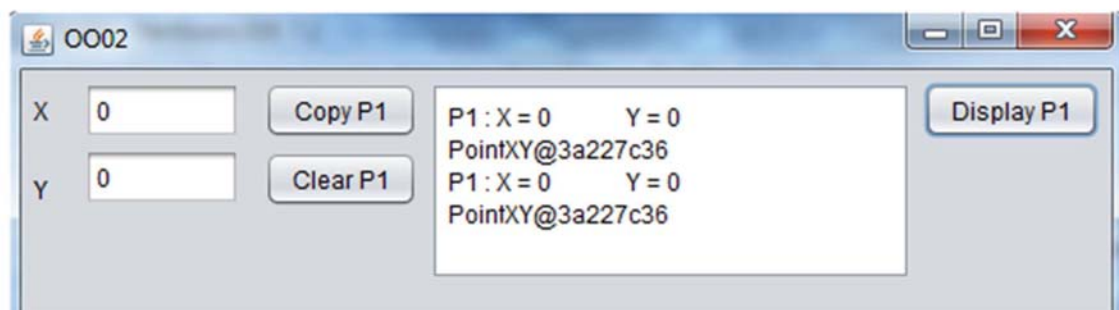
```
1  +  /*...*/  
5  
6  
7  +  /**...*/  
11 public class PointXY {  
12     int X;  
13     int Y;  
14     public void initPointXY()  
15     {  
16         X=0;  
17         Y=0;  
18     }  
19 }  
20
```

- 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
```

## 0003

# Check Lab manual



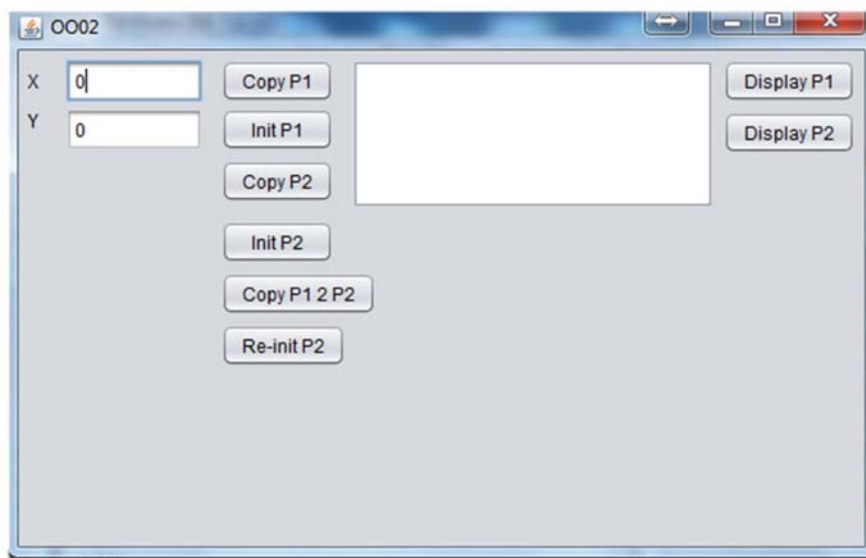
- 
- A variable you have declared of a class is also called an instance of the class.
  - In the same way, you can declare various instances of the same class as necessary:
  - Each one of these instances gives you access to the members of the class but each instance holds the particular values of the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

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

## 0004

---

## 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.

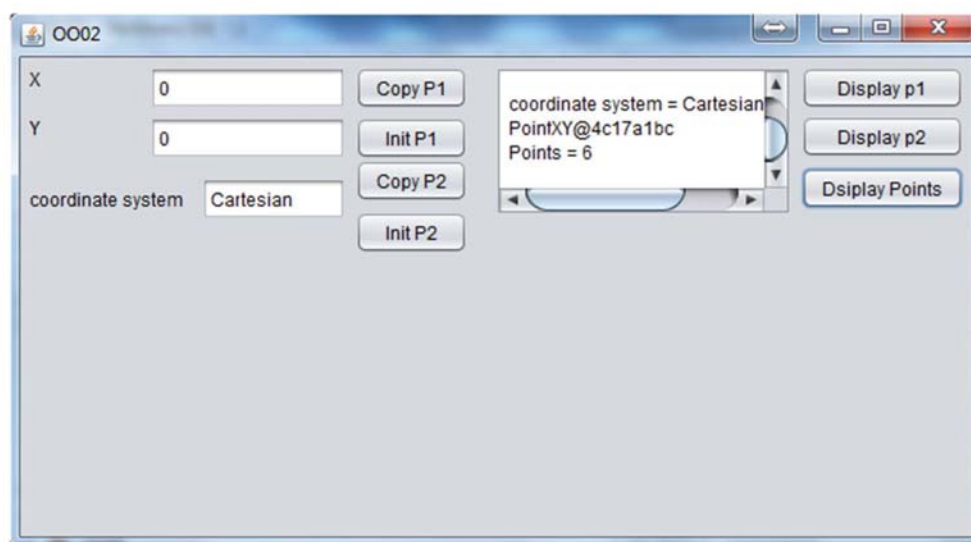
```
1  /*...*/
5
6
7  /*...*/
11 public class PointXY {
12     static String coordinateType;
13     int X;
14     int Y;
15     public void initPointXY()
16     {
17         X=0;
18         Y=0;
19     }
20 }
21
```

١٩

Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013

## 0005

## Check Lab manual



٢٠

Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013

# 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 }
```

٢١

Dr. Ahmed ElShafee, Fundamen 21  
ACU/CSIT Fall 2013

## Characteristics of Members of a Class

### Constants

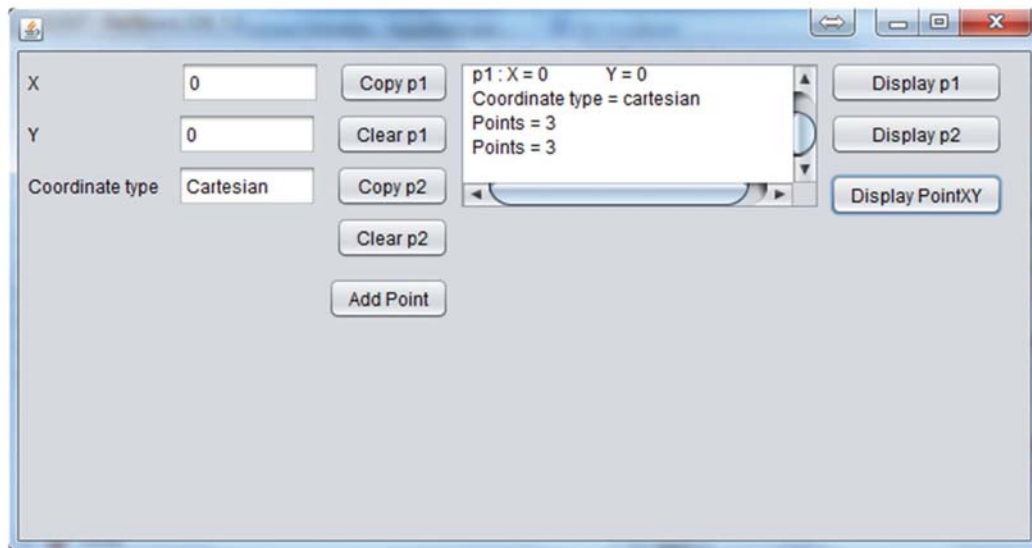
- You can create a constant variable in a class. To create a constant variable, type the **final** keyword to the left of the variable.
- Once again, when declaring a constant, you must initialize it with an appropriate constant value.

```
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
```

٢٢

Dr. Ahmed El

## 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
```

٢٥

Dr. Ahmed ElShafi

0008

---

**Check lab manual**

X: 2  
Y: 2  
Coordinate type: Cartesian

Copy p1  
Clear p1  
Copy p2  
Clear p2

p1: X = 1 Y = 1  
Coordinate type = cartesian  
p2: X = 2 Y = 2  
Coordinate type = cartesian

Display p1  
Display p2

٢٦

## 0009 (Class return methods)

**Check lab manual**

X: 2  
Y: 2  
Coordinate type: Cartesian

Copy p1  
Clear p1  
Copy p2  
Clear p2

p1 : X = 1    Y = 1  
Coordinate type = Cartesian  
p2 : X = 2    Y = 2  
Coordinate type = Cartesian

Display p1  
Display p2

٢٧

Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013

## 0010 (Methods' Arguments)

**Check lab manual**

X: 6  
Y: 7  
Coordinate type: Cartesian

Copy p1  
Clear p1  
Copy p2  
Clear p2

p1 : X = 3    Y = 4  
Coordinate type = Cartesian  
p2 : X = 6    Y = 7  
Coordinate type = Cartesian

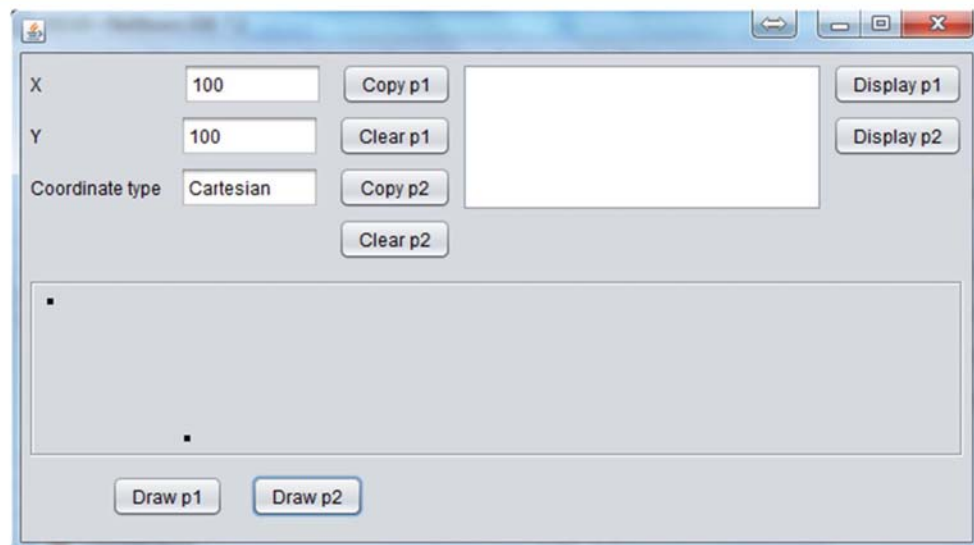
Display p1  
Display p2

٢٨

Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013

# 0011

## Check lab manual

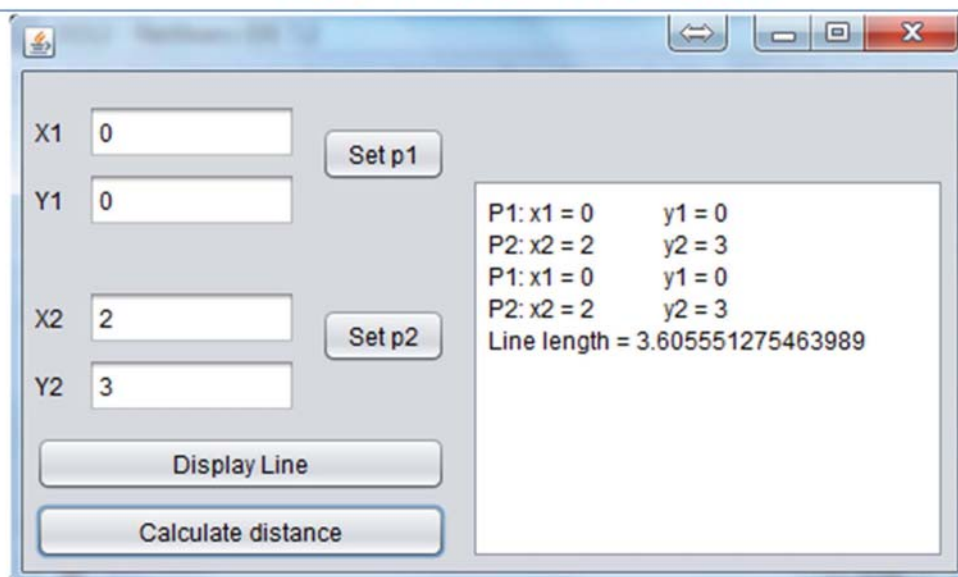


٢٩

Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013

# 0012

## Check lab manual



٣٠

Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013

# Assignments 1

A Java Swing window titled "Triangle Calculator" with a standard Windows-style title bar. The window contains several input fields and buttons. On the left, there are six input fields labeled X1, Y1, X2, Y2, X3, and Y3, each with a corresponding "Set" button (Set p1, Set p2, Set p3). In the center, there is a text area displaying the coordinates of the triangle's vertices (P1, P2, P3) and the calculated perimeter and area. On the right, there are three buttons: "Display Triangle", "Calculate Perimeter", and "Calculate Area".

| Point | X | Y |
|-------|---|---|
| P1    | 1 | 1 |
| P2    | 2 | 2 |
| P3    | 3 | 3 |

Triangle Perimeter = 2.8284271  
Triangle Area = 4.0

٣١

Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013

# Assignment 2

A Java Swing window titled "مشروع كشك جرائد" (Newspaper Kiosk Project) with a standard Windows-style title bar. The window contains a list of items with their respective costs and checkboxes for selection. The items are: أخبار (98), أهرام (98), المصري اليوم (48), الدستور (48), and التحرير (49). Below the list, there are two buttons: "تقرير" (Report) and "بيع" (Sell). At the bottom, there is a text area displaying the selected items and their costs.

| Item         | Cost | Selected                 |
|--------------|------|--------------------------|
| أخبار        | 98   | <input type="checkbox"/> |
| أهرام        | 98   | <input type="checkbox"/> |
| المصري اليوم | 48   | <input type="checkbox"/> |
| الدستور      | 48   | <input type="checkbox"/> |
| التحرير      | 49   | <input type="checkbox"/> |

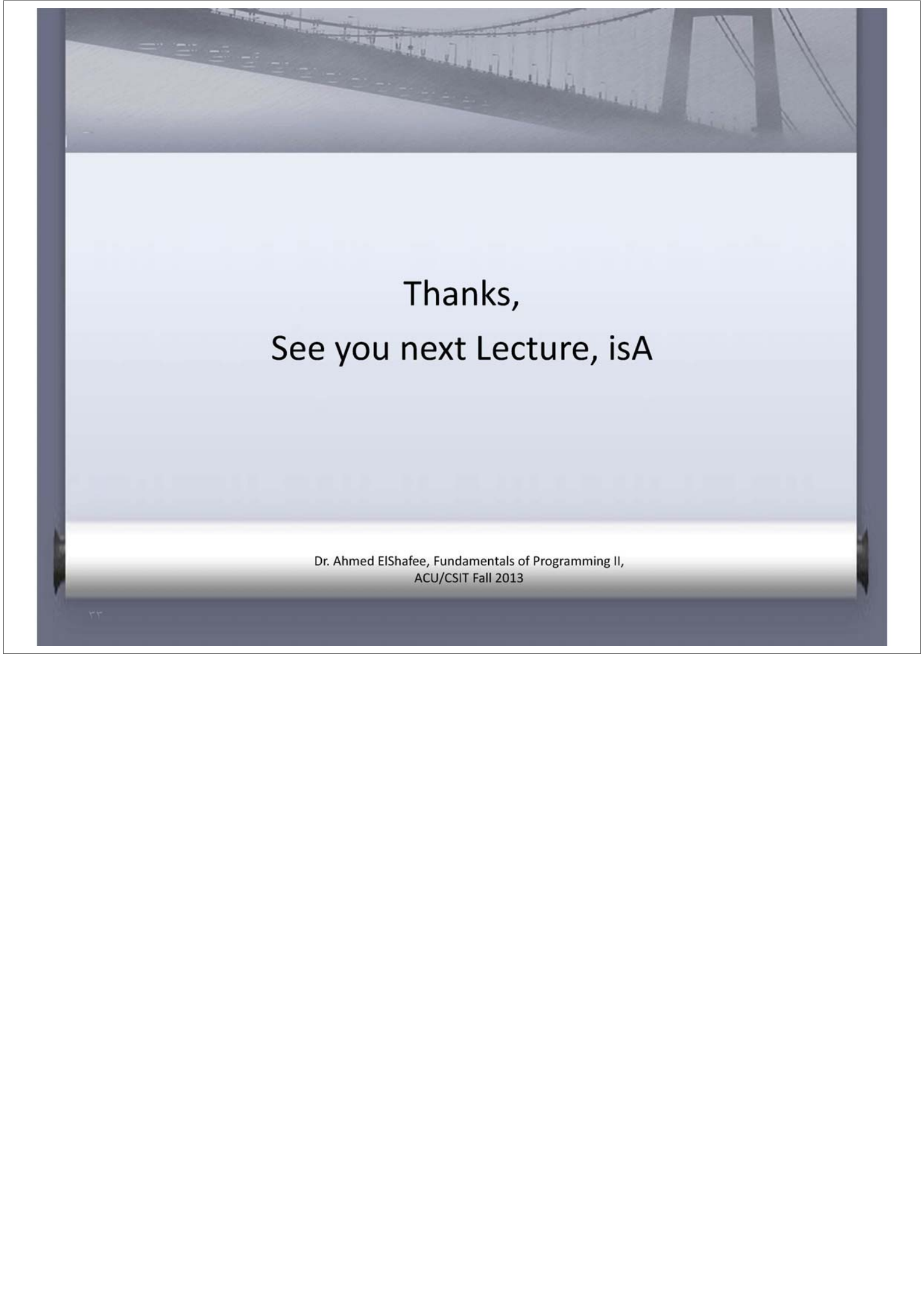
تقرير بيع

98: أخبار  
98: أهرام  
48: المصري اليوم  
48: الدستور  
49: التحرير

٣٢

Dr. Ah

ACU/CSIT Fall 2013

The background of the slide is a photograph of a suspension bridge, likely the Bix Creek Creek Bridge in the United States, featuring a large concrete arch and steel cables. The image is slightly faded and serves as a backdrop for the text.

Thanks,  
See you next Lecture, isA

Dr. Ahmed ElShafee, Fundamentals of Programming II,  
ACU/CSIT Fall 2013