

**King Saud University**  
**College of Computer & Information Science**  
**CSC111 – Tutorial09**  
**Object – I –**  
**All Sections**

---

### **Objectives:**

- To describe objects and classes, and use classes to model objects.
- To use UML graphical notation to describe classes and objects.
- To demonstrate how to define classes and create objects.
- To create objects using constructors.
- To access objects via object reference variables.
- To define a reference variable using a reference type.
- To access an object's data and methods using the object member access operator (.).
- To define data fields of reference types and assign default values for an object's data fields.
- To distinguish between object reference variables and primitive data type variables.

### **Exercise 1**

- 1) What is wrong with the following program (Note: When you write a class, you can put the `main` method inside that class to quickly test the class without writing two separate classes (i.e., instead of writing a new "Test" class along with the current class). In general, it is better to separate the class that contains `main` method from other classes in your program):

```
1 public class ShowErrors {  
2     public static void main(String[] args) {  
3         ShowErrors t = new ShowErrors(5);  
4     }  
5 }
```

(a)

```
1 public class ShowErrors {  
2     public static void main(String[] args) {  
3         ShowErrors t = new ShowErrors();  
4         t.x();  
5     }  
6 }
```

(b)

```
1 public class ShowErrors {  
2     public void method1() {  
3         Circle c;  
4         System.out.println("What is radius "  
5             + c.getRadius());  
6         c = new Circle();  
7     }  
8 }
```

(c)

```

1  public class ShowErrors {
2      public static void main(String[] args) {
3          C c = new C(5.0);
4          System.out.println(c.value);
5      }
6  }
7
8  class C {
9      int value = 2;
10 }

```

(d)

2) Identify and correct the errors in the following program:

```

1  class Test {
2      public static void main(String[] args) {
3          A a = new A();
4          a.print();
5      }
6  }
7
8  class A {
9      String s;
10
11     A(String newS) {
12         s = newS;
13     }
14
15     public void print() {
16         System.out.print(s);
17     }
18 }

```

## Solution

1)

```
1 public class ShowErrors {
2     public static void main(String[] args) {
3         ShowErrors t = new ShowErrors(5);
4     }
5 }
```

Class ShowErrors does not have a constructor that accepts an integer parameter

(a)

```
1 public class ShowErrors {
2     public static void main(String[] args) {
3         ShowErrors t = new ShowErrors();
4         t.x();
5     }
6 }
```

Class ShowErrors does not have a field named x.

(b)

```
1 public class ShowErrors {
2     public void method1() {
3         Circle c;
4         System.out.println("What is radius "
5             + c.getRadius());
6         c = new Circle();
7     }
8 }
```

Class Circle is not defined

(c)

```

1  public class ShowErrors {
2      public static void main(String[] args) {
3          C c = new C(5.0);
4          System.out.println(c.value);
5      }
6  }
7
8  class C {
9      int value = 2;
10 }

```

Class C does not have a constructor that accepts an argument

(d)

2) public is missing

```

1  class Test {
2      public static void main(String[] args) {
3          A a = new A();
4          a.print();
5      }
6  }
7
8  class A {
9      String s;
10
11     A(String newS) {
12         s = newS;
13     }
14
15     public void print() {
16         System.out.print(s);
17     }
18 }

```

Class A does not have a default constructor (constructor with no parameters). You need to pass a String argument here.

## Exercise 2

Design a class named **Ball** class that models a moving ball. The class contains:

- Two properties **x**, **y** which maintain the position of the ball in a two dimensional space.
- A default constructor that sets position to (0, 0).
- A constructor that receives two parameters x and y that represent current position of the ball.
- Methods **getX** and **getY** that return the current position of the ball.
- A method **setPos**, which sets the position of the ball.
- A method **move**, which changes x and y by the given **xDisp** and **yDisp**, respectively.
- A method **toString**, which returns the string "**Ball @ (x,y)**".

Start by drawing the UML for the class Ball. Then write a program that does the following:

- It creates a new ball with a position read from the user.
- Then it moves the ball by (3, -5).
- Finally, it prints the new position of the balls using **toString** method of class Ball.

## Solution

Ball
x: double y: double
Ball() Ball(newX: double, newY: double) getX(): double getY(): double setPos(newX: double, newY: double): void move(xDisp: double, yDisp: double): void toString(): String

TestBall
main(): void

```
public class Ball {  
    // data members (properties)  
    double x, y; // x and y location  
  
    // Constructors  
    public Ball(double newX, double newY) {  
        x = newX;  
        y = newY;  
    }  
    public Ball() {  
        x = 0.0;  
        y = 0.0;  
    }  
  
    // Getters fields x and y  
    public double getX() {  
        return x;  
    }  
    public double getY() {  
        return y;  
    }  
  
    public void setPos(double newX, double newY) {  
        x = newX;  
        y = newY;  
    }  
  
    public void move(double xDisp, double yDisp) {  
        x += xDisp;  
        y += yDisp;  
    }  
  
    public String toString() {  
        return "Ball @ (" + x + "," + y + ")";  
    }  
}
```



```
import java.util.Scanner;
public class TestBall {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter ball positions (x, y): ");
        double x = input.nextDouble();
        double y = input.nextDouble();
        Ball ball = new Ball(x, y);
        ball.move(3, -5);
        System.out.println(ball.toString());
    }
}
```

**Done...**