**King Saud University**
**College of Computer & Information Science**
**CSC111 – Tutorial10**
**Object – II –**
**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. Suppose that the class F is defined in (a). Let f be an instance of F.

   Which of the statements in (b) are correct?

```
public class F {
  int i;
  static String s;

  void imethod() {
  }

  static void smethod() {
  }
}
```

(a)

```
System.out.println(f.i);
System.out.println(f.s);
f.imethod();
f.smethod();
System.out.println(F.i);
System.out.println(F.s);
F.imethod();
F.smethod();
```

(b)

2. Add static keyword in place of ? if appropriate

```
public class Test {
  int count;

  public ? void main(String[] args) {
    ...
  }

  public ? int getCount() {
    return count;
  }

  public ? int factorial(int n) {
    int result = 1;
    for (int i = 1; i <= n; i++)
      result *= i;

    return result;
  }
}
```

3. In each place where there is a **?**, list all properties of class C1 that are accessible and the ones that are not accessible. Also list all methods that can be invoked and the ones that cannot be invoked.

```
package p1;

public class C1 {
  public int x;
  int y;
  private int z;

  public void m1() {
  }
  void m2() {
  }
  private void m3() {
  }
}
```

```
package p1;

public class C2 {
  void aMethod() {
    C1 o = new C1();

    ?

  }
}
```

```
package p2;

public class C3 {
  void aMethod() {
    C1 o = new C1();

    ?

  }
}
```

4. Put a line under the errors in the following program (Notice: the program consists of two files):

```
public class FullOfErrors {
        private int prop1 ;
        private double prop2;
        public FullOfErrors(int p1){
            prop1 = p1;
        }
        public void setProp1(double p){
            prop1 = p;
        }
        public int getProp2(){
            return prop2;
        }
        public int getProp1(){
            System.out.println("prop1= "+prop1);
        }
        public void setProp1Prop2(double a, int b){
            prop1 = b;  prop2 = a;
        }
}
```

```
public class TestFullOfErrors {
        public static void main(String[] args) {
            FullOfErrors m = new FullOfErrors();
            FullOfErrors m2 = FullOfErrors(5);
            int x = 1;  int y;
            y = m.setProp1(x + 3);
            m.setProp1Prop2(1, 1.0);
            m.prop2 = 2.0;
        }

}
```

5. What is the output of the following program?

```
class Magic {
    int i;
    double j;
}
public class TestMagic {
    public static void main(String[] args) {
        Magic m = new Magic();
        m.i = 11;
        m.j = 5.5;
        Magic m2 = new Magic();
        m2 = m;
        m2.i = m2.i + 2;
        m2.j = 1 + m2.i / (( m.i - 9) / 2);
        System.out.println(m.i + ", " + m.j +", " + m2.i + ", " + m2.j);
    }
}
```

## Solution

1)

```
public class F {
  int i;
  static String s;

  void imethod() {
  }

  static void smethod() {
  }
}
```

(a)

```
System.out.println(f.i);
System.out.println(f.s);
f.imethod();
f.smethod();
System.out.println(F.i);
System.out.println(F.s);
F.imethod();
F.smethod();
```

(b)

2)

```
public class Test {
  int count;

  public void main(String[] args) {
    ...
  }

  public int getCount() {
    return count;
  }

  public int factorial(int n) {
    int result = 1;
    for (int i = 1; i <= n; i++)
      result *= i;

    return result;
  }
}
```

3)

```java
package p1;

public class C1 {
  public int x;
  int y;
  private int z;

  public void m1() {
  }
  void m2() {
  }
  private void m3() {
  }
}
```

```java
package p1;

public class C2 {
  void aMethod() {
    C1 o = new C1();
    can access o.x;
    can access o.y;
    cannot access o.z;

    can invoke o.m1();
    can invoke o.m2();
    cannot invoke o.m3();
  }
}
```

```java
package p2;

public class C3 {
  void aMethod() {
    C1 o = new C1();
    can access o.x;
    cannot access o.y;
    cannot access o.z;

    can invoke o.m1();
    cannot invoke o.m2();
    cannot invoke o.m3();
  }
}
```

4)

```java
public class FullOfErrors {
        private int prop1 ;
        private double prop2;
        public FullOfErrors(int p1){
            prop1 = p1;
        }
        public void setProp1(double p){
            prop1 = p;
        }
        public int getProp2(){
            return prop2;
        }
        public int getProp1(){
            System.out.println("prop1= "+prop1);
        }
        public void setProp1Prop2(double a, int b){
            prop1 = b;  prop2 = a;
        }
}
```

```
3  public class TestFullOfErrors {
4          public static void main(String[] args) {
5              FullOfErrors m = new FullOfErrors();
6              FullOfErrors m2 = FullOfErrors(5);
7              int x = 1;  int y;
8              y = m.setProp1(x + 3);
9              m.setProp1Prop2(1, 1.0);
10             m.prop2 = 2.0;
11         }
12
13 }
```

5)

**OUTPUT**

```
13, 7.0, 13, 7.0
```

## Exercise 2

Design a class named **Fan** to represent a fan. The class contains:

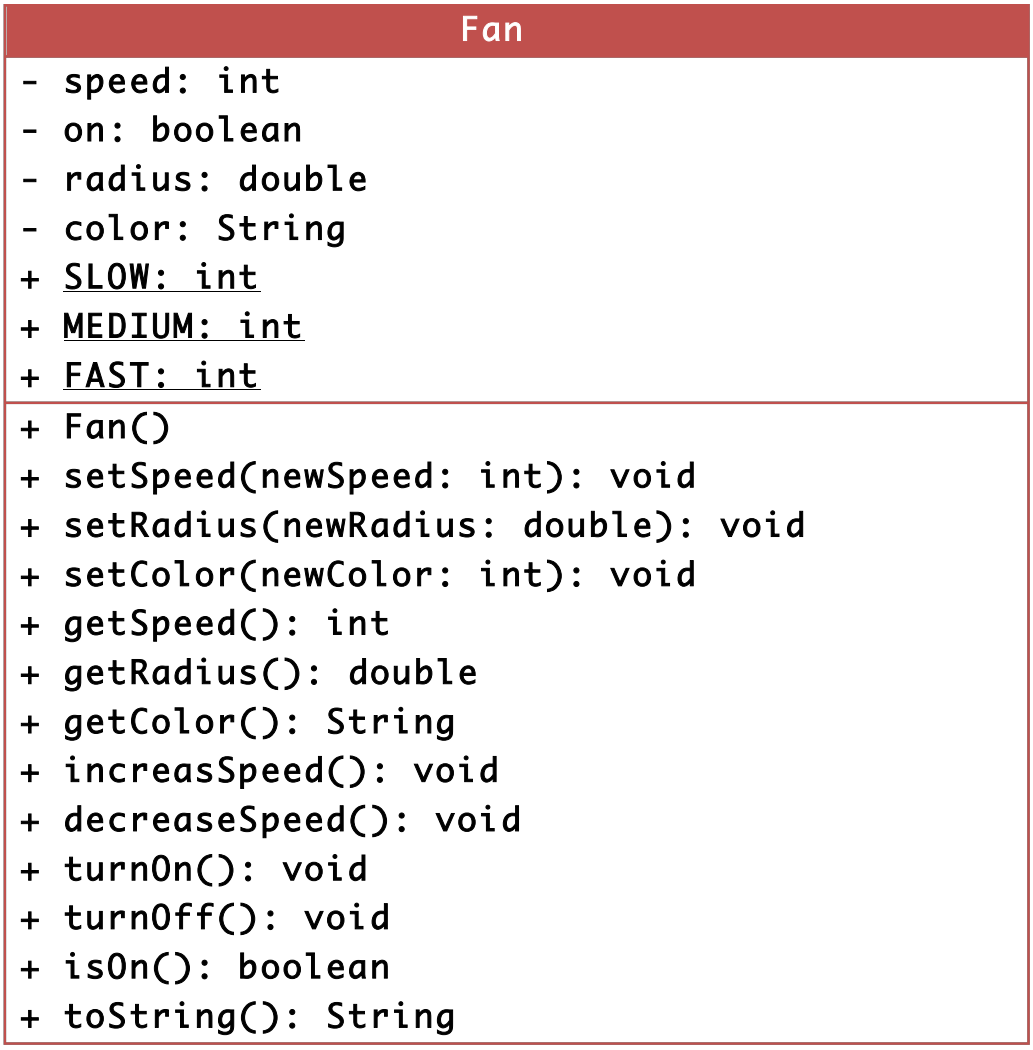
- Three constants named **SLOW**, **MEDIUM**, and **FAST** with the values **1**, **2**, and **3** to denote the fan speed.
- A private **int** data field named **speed** that specifies the speed of the fan (the default is **SLOW**).
- A private **boolean** data field named **on** that specifies whether the fan is on (the default is **false**).
- A private **double** data field named **radius** that specifies the radius of the fan (the default is **5**).
- A string data field named **color** that specifies the color of the fan (the default is **blue**).
- The accessor (getter) and mutator (setter) methods for all data fields except **on**.
- A no-arg constructor that creates a default fan.
- A method **turnOn()** to turn on the fan.
- A method **turnOff()** to turn off the fan.
- A method **increaseSpeed()** that increases the speed of the fan unless the fan is running at highest speed. If it is running at highest speed then the method prints an error message "**Fan is already running at highest speed.**".
- A method **deceaseSpeed()** that decreases the speed of the fan unless the fan is running at lowest speed. If it is running at loweset speed then the method prints an error message "**Fan is already running at lowest speed.**".

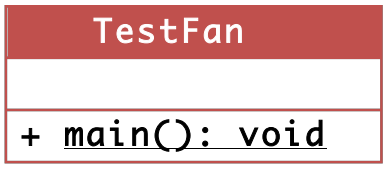
- A method named **toString()** that returns a string description for the fan. If the fan is on, the method returns the fan speed, color, and radius in one combined string. If the fan is not on, the method returns the fan color and radius along with the string "`fan is off`" in one combined string.

Draw the UML diagram for the class and then implement the class. Write a test program that doest the following:

- Creates two **Fan** objects.
- Assigns maximum speed, radius **10**, color **yellow** to the first object.
- Turns first object on.
- After first object is turned on, it increases its speed.
- Assigns medium speed, radius **5**, color **blue** to the second object.
- Turns second object on.
- Then decreases its speed twice,
- After that it turns it off.
- Displays the two objects by invoking their **toString()** method.

## Solution


Fan
- speed: int
- on: boolean
- radius: double
- color: String
+ SLOW: int
+ MEDIUM: int
+ FAST: int
+ Fan()
+ setSpeed(newSpeed: int): void
+ setRadius(newRadius: double): void
+ setColor(newColor: int): void
+ getSpeed(): int
+ getRadius(): double
+ getColor(): String
+ increaseSpeed(): void
+ decreaseSpeed(): void
+ turnOn(): void
+ turnOff(): void
+ isOn(): boolean
+ toString(): String


TestFan
+ main(): void

```
class Fan {
    public static int SLOW = 1;
    public static int MEDIUM = 2;
    public static int FAST = 3;

    private int speed = SLOW;
    private boolean on = false;
    private double radius = 5;
    private String color = "white";

    public Fan() {
    }

    public void increaseSpeed() {
        if (speed == SLOW)
            speed = MEDIUM;
        else if (speed == MEDIUM)
            speed = FAST;
        else
            System.out.println("Fan is already running at highest speed.");
    }

    public void decreaseSpeed() {
        if (speed == FAST)
            speed = MEDIUM;
        else if (speed == MEDIUM)
            speed = SLOW;
        else
            System.out.println("Fan is already running at lowest speed.");
    }

    public int getSpeed() {
        return speed;
    }

    public void setSpeed(int newSpeed) {
        speed = newSpeed;
    }

    public boolean isOn() {
        return on;
    }

    public void turnOn() {
        on = true;
    }
```

```
    public void turnOff() {
        on = false;
    }

    public double getRadius() {
        return radius;
    }

    public void setRadius(double newRadius) {
        radius = newRadius;
    }

    public String getColor() {
        return color;
    }

    public void setColor(String newColor) {
        color = newColor;
    }

    @Override
    public String toString() {
        String result = "speed " + speed + "\n" + "color " + color + "\n" + "radius "
                + radius + "\n";
        if (isOn())
            result += "fan is on";
        else
            result += " fan is off";
        return result;
    }
}
```

```java
public class TestFan {
    public static void main(String[] args) {
        Fan fan1 = new Fan();
        fan1.setSpeed(Fan.FAST);
        fan1.setRadius(10);
        fan1.setColor("yellow");
        fan1.turnOn();
        fan1.increaseSpeed();
        System.out.println(fan1.toString());

        Fan fan2 = new Fan();
        fan2.setSpeed(Fan.MEDIUM);
        fan2.setRadius(5);
        fan2.setColor("blue");
        fan2.turnOn();
        fan2.decreaseSpeed();
        fan2.decreaseSpeed();
        fan2.turnOff();
        System.out.println(fan2.toString());
    }
}
```

**Done…**