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

Suppose that the class F is defined in (a). Let f be an instance of F.
 Which of the statements in (b) are correct?

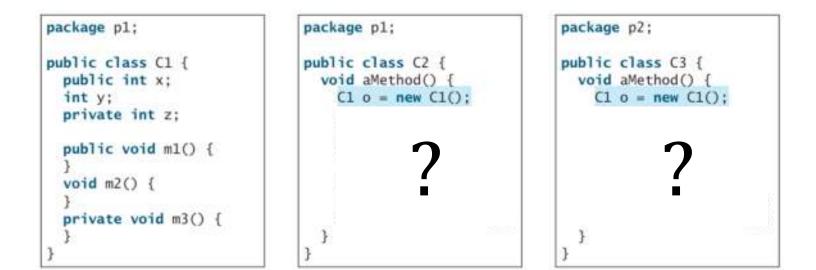
```
public class F {
    int i;
    static String s;
    void imethod() {
    }
    static void smethod() {
    }
}
```

```
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();
```

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;
    }
}</pre>
```

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.



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;
        3
        public void setProp1(double p){
            prop1 = p;
        3
        public int getProp2(){
            return prop2;
        3
        public int getProp1(){
            System.out.println("prop1= "+prop1);
        3
        public void setProp1Prop2(double a, int b){
            prop1 = b; prop2 = a;
        3
```

```
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();

System.out.println(F.s);

F.smethod();

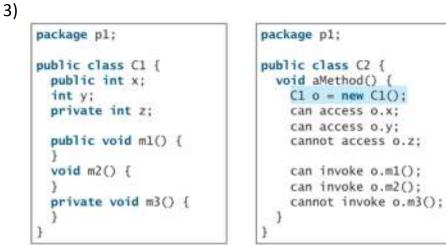
System.out.println(F.s);

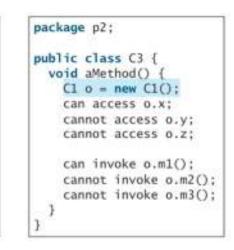
System.out.println(F.s);
```

(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;
    }
}</pre>
```





4)

3 publi	c class FullOfErrors {
4	private int prop1 ;
5	private double prop2;
60	<pre>public FullOfErrors(int p1){</pre>
7	prop1 = p1;
8	}
90	<pre>public void setProp1(double p){</pre>
10	prop1 = p;
11	}
12 <del>0</del>	<pre>public int getProp2(){</pre>
213	return prop2;
14	}
2150	<pre>public int getProp1(){</pre>
16	<pre>System.out.println("prop1= "+prop1);</pre>
17	}
180	<pre>public void setProp1Prop2(double a, int b){</pre>
19	prop1 = b; prop2 = a;
20	}
21 }	

	3	<pre>public class TestFullOfErrors {</pre>	
	49	<pre>public static void main(String[] args</pre>	;) {
<b>B</b>	5	FullOfErrors m = new FullOfErrors	;O;
10	6	FullOfErrors $m2 = FullOfErrors(5)$	);
	7	int x = 1; int y;	
1	8	y = m.setProp1(x + 3);	
<b>N</b> B	9	m.setProp1Prop2(1, 1.0);	
2	10	m.prop2 = 2.0;	
	11	}	
	12		
	13	}	

# <sup>5)</sup> **OUTPUT**

13, 7.0, 13, 7.0

Done...