

## Part A. Class & Objects

1. What is class and how is it different from object?

Class is a definition for the object that were created by it, which class define types of data and operations of an object.

2. What are the components of a class?

- Constructor
- Data attributes
- Methods

3. How many types of classes are there? Explain

Driver Class - Class that runs a program (contain main method)

Boundary Class (Service, Utility) - collects the related static methods - no data

Entity Class - represent real world objects

4. The method `multiply` has the following properties:

- Cannot be invoked from outside of the class
- Receive two integers x and y
- Returns the product of x and y

Write the code for the above method

```
private int multiply (int x, int y) {  
    return x * y;  
}
```

bib

5. What is the output of the following program?

$$1 \xrightarrow{x++} 2 \xrightarrow{x--} 1 \xrightarrow{x^2} 2 \xrightarrow{x^2} 4 \xrightarrow{x^2} 8$$

```

1 class Magic {
2     public int x = 1;
3
4     public void bibbidi() {
5         x++;
6         bobbidi();
7         boo();
8     }
9
10    public void bobbidi() {
11        x--;
12        boo();
13        boo();
14    }
15
16    public void boo() {
17        x *= 2;
18    }
19 }
20
21 public class Main {
22     public static void main(String[] args) {
23         Magic m = new Magic();
24         m.bibbidi();
25         m.bobbidi();
26         m.boo();
27         System.out.println(m.x);
28     }
29 }
```

bob

$$\begin{aligned} 8 &\xrightarrow{x--} 7 \\ 7 &\xrightarrow{x^2} 14 \\ 14 &\xrightarrow{x^2} 28 \end{aligned}$$

boo

$$28 \xrightarrow{x^2} \boxed{56}$$

#

6. What happens if we add keyword **static** to a variable or method?

static variable  $\Rightarrow$  there can be only one copy of "it"  
which will be allocated memory  
when class first referenced.

static method  $\Rightarrow$  "it" belongs to the class  
and can be called directly from class  
without object creation

7. What is the output of the following program?

10 (default ×) #

```
class Point {  
    int m_x, m_y;  
  
    public Point(int x, int y) { m_x = x; m_y = y; }  
    public Point() { this(10, 10); }  
    public int getX() { return m_x; }  
    public int getY() { return m_y; }  
  
    public static void main(String args[]) {  
        Point p = new Point();  
        System.out.println(p.getX());  
    }  
}
```

8. What is the output of the following program?

```
class Complex {  
    private double re, im;  
  
    public Complex(double re, double im) {  
        this.re = re;  
        this.im = im;  
    }  
  
    Complex(Complex c)  
    {  
        System.out.println("Copy constructor called");  
        re = c.re;  
        im = c.im;  
    }  
  
    public String toString() {  
        return "(" + re + " + " + im + "i)";  
    }  
}  
class Main {  
    public static void main(String[] args) {  
        Complex c1 = new Complex(10, 15);  
        Complex c2 = new Complex(c1);  
        Complex c3 = c1;  
        System.out.println(c2);  
    }  
}
```

Deprecated

Copy constructor called  
( 10.0 + 15.0i )

#

## Part B. OOP

1. Explain each concept of Object-Oriented Programming, what it is, how to achieve it in Java, why it is useful.

Inheritance - based object, class upon other object, class  
that have similar implementation

Encapsulation - Hiding data, exposed via specific interfaces

Abstraction - Hiding implementation, shown only feature

Polymorphism - Object can take many form

2. What is Object-Oriented Programming? What are the advantages?

dealing with "Object" that bind data and functions to operate on them.

Advantage

- Breaks into sub-problem (one object at a time)
- code reusability, scalable
- by Inheritance, reduced redundant code

3. Explain OOP in one sentence

Programming paradigm which treats data and procedure as "object"

4. How is encapsulation different from abstraction?

-----

-----

-----