**1.Given the following code, select the correct output, if any:**

class Flower {

public void fragrance() {

System.out.println("Flower"); }

}

class Rose {

public void fragrance() {

System.out.println("Rose"); }

}

class Lily {

public void fragrance() {

System.out.println("Lily"); }

}

class Bouquet {

public void arrangeFlowers() {

Flower f1 = new Rose();

Flower f2 = new Lily();

f1.fragrance(); }

}

1. Flower
2. Rose
3. Lily
4. Compilation error

**Explanation**

Although the code seems to implement polymorphism using classes, note that neither of the classes Rose or Lily extends the class Flower. Hence, a variable of type Flower can’t be used to store objects of the classes Rose or Lily.

**2.What is the output of the following code?**

public class Excel{

public static void main(String[] args) {

int x[][] = new int[4][];

x[0] = new int[]{2, 5, 8};

x[1] = new int[]{1, 3};

system.out.println(x[0][1]);

}

}

1. 2
2. 5
3. 25
4. Compilation error

**Explanation** : x[][] is a two dimensional array. We have assigned two one dimensional int arrays x[0] and x[1] into the two dimensional array x. So the code compiles fine and produces the output as 5.

3. What is the output for this code ?

**public static void** main(String[] args) {  
 String [] mango = {**"Red Mango"**, **"Yellow Mango"**};  
 System.***out***.println( mango.**length**);  
 **for** (**int** i = 0; i < mango.**length**; i++) {  
 System.***out***.print(mango[i] + **","**);  
 **break**;  
 }  
}

1. Red Mango, Yellow Mango,
2. Red Mango,
3. Yellow Mango,
4. Compilation error

**Explanation**

The first time through the loop, the index is 0 and Red Mango, is output. The break statement (line 5) then skips all remaining executions on the loop and the main() method ends.

**4. What will happen when you compile and run the following code?**

**class** Numbers {  
 **private int max** = 10;  
 **public static void** main(String[] args) {  
 **int** min = 20;  
 System.***out***.println(min + **max**);  
 }  
}

1. 20
2. 30
3. Compilation error
4. Min

Explanation :

You cannot access the instance variable "m" from a static method

5. The output of the **class** TestEJavaCourse, defined as follows, is 300:  
**class** Course {  
 **int enrollments**;  
}  
  
**class** TestEJavaCourse {  
 **public static void** main(String args[]) {  
 Course c1 = **new** Course();  
 Course c2 = **new** Course();  
 c1.**enrollments** = 100;  
 c2.**enrollments** = 200;  
 System.***out***.println(c1.**enrollments** + c2.**enrollments**);  
 }  
}  
What is the output **if** the variable enrollments is defined as a **static** variable?

1. Compilation error
2. 300
3. 400
4. 500

Explanation

The code doesn’t fail compilation after the definition of the variable enrollments is changed to a static variable. A static variable can be accessed using a variable reference of the class in which it’s defined. All the objects of a class share the same copy of the static variable. When the variable enrollments is modified using the reference variable c2, c1.enrollments is also equal to 200. Hence, the code prints the result of 200 + 200, that is, 400.

6. What code in the dash \_\_\_\_\_ will enable the **class** Jungle to determine whether the reference variable animal refers to an object of the **class** Lion and print 1?  
  
**class** Animal {  
 **float age**;  
}  
  
**class** Lion **extends** Animal {  
 **int claws**;  
}  
  
**class** Jungle {  
  
 **public static void** main(String args[]) {  
 Animal animal = **new** Lion();  
  
 **if** (\_\_\_\_\_\_\_\_\_\_\_\_)  
  
 System.***out***.println(1);  
 }  
  
}

1. animal instanceOf Lion
2. animal instanceof Lion
3. animal==Lion
4. animal=Lion

Explaination :

The correct operator name is instanceof and not instanceOf (note the capitalized O). Last two options are incorrect. Neither of these lines of code will compile because they are trying to compare and assign a class name to a variable, which isn’t allowed.

7. **Given the following code, which option, if used to replace /\* INSERT CODE HERE \*/, will cause the code to print 110?**

**class** Book {  
 **private int pages** = 100;  
}  
  
**class** Magazine **extends** Book {  
 **private int interviews** = 2;  
 **private int** totalPages() {  
 */\* INSERT CODE HERE \*/* }  
  
 **public static void** main(String[] args) {  
 System.***out***.println(**new** Magazine().totalPages());  
 }  
}

1. return super.pages+this.interviews\*5
2. return this.pages+interviews\*5
3. return this.pages+super.interviews\*5
4. None of the above

Explaination :

The variable pages has private access in the class Book, and it can’t be accessed from outside this class.

8.