Java SE 8. - Test 2

1.Given

```
class Book {
	int id;
	String name;

	public Book (int id, String name) {
		this.id = id;
		this.name = name;
	}
	public boolean equals (Object obj) { //line n1
		boolean output = false;
		Book b = (Book) obj;
		if (this.name.equals(b.name)) {
			output = true;
		}
		return output;
	}
}
```

and the code fragment:

```
Book b1 = new Book (101, "Java Programing");
Book b2 = new Book (102, "Java Programing");
System.out.println (b1.equals(b2)); //line n2
```

Which statement is true?

A. The program prints true

B. The program prints false

C. A compilation error occur in line 1

D. To ensure successful compilation, replace line n1 with: boolean equals (Book obj) {

E. A compilation error occur in line 2

F. To ensure successful compilation, replace line n2 with: System.out.println (b1.equals((Object) b2));

2. Given:

```
public class Customer {
	private String fName;
	private String lName;
	private static int count;

	public Customer (String first, String last) {
		fName = first;
		lName = last;
		++count;
	}
	static { count = 0; }
	public static int getCount() {return count; }
}
```

```
public class App {

        public static void main (String [] args) {
                Customer c1 = new Customer("Larry", "Smith");
                Customer c2 = new Customer("Pedro", "Gonzales");
                Customer c3 = new Customer("Penny", "Jones");
                Customer c4 = new Customer("Lars", "Svenson");
                c4 = null;
                c3 = c2;
                System.out.println (Customer.getCount());
        }
}
```

What is the result?
A. 2
B. 3
C. 4
D. 5

---

3. Given:

```
abstract class Shape { //line 1
        Shape ( ) {
                System.out.println ("Shape");
        }
        protected void area () {
                System.out.println ("Shape");
        }
}

class Square extends Shape {

        int side;
        Square (int side) {
                //line 2
        }
        public  void area () {  //line 3
                System.out.println ("Square");
        }

}


class Rectangle extends Square{
        int len, br;
        Rectangle(int x, int y){
                //line 4
                len = x;
                br = y;
        }
        void area ( ) {  //line 5
                System.out.println ("Rectangle");
  }
```

}

Which two modifications enable the code to compile?

A. At line 1, remove abstract
B. At line 2, insert super ( );
C. At line 3, remove public
D. At line 4, insert super (x);
E. At line 4, insert super (); super.side = x;
F. At line 5, use public void area ( ) {

---

4. Given:

```
interface Downloadable{
    public void download();
}
interface Readable extends Downloadable {  // line n1
    public void readBook();
}
abstract class Book implements Readable { // line 2
    public void readBook(){
        System.out.println("Read Book");
    }
}
class EBook extends Book{  // line n3
    public void readBook(){
        System.out.println("Read E-Book");
    }
}
```

And given the code fragment:

```
Book book1 = new EBook();
book1.readBook();
```

What is the result?

A. Compilation fails at line n2.
B. Read E-Book
C. Compilation fails at line n1.
D. Compilation fails at line n3.
E. Read Book

---

5. Given:

```
class Student{
    String name;

    public Student(String name){
        this.name = name;
    }
}
public class Test {
    public static void main(String[] args) {
        Student[] students = new Student[3];
        students[1] = new Student("Richard");
        students[2] = new Student("Donald");
        for (Student s : students){
            System.out.println("" + s.name);
        }
    }
}
```

What is the result?

A. An ArrayIndexOutOfBoundsException is thrown at runtime.
B. Richard Donald
C. A NullPointerException is thrown at runtime.
D. Compilation fails.
E. null Richard Donald

---

6. ¿Qué tres sentencias describen las características de la programación orientada a objetos del lenguaje Java?

A. Se debe declarar un método main en cada clase
B. Los objetos pueden compartir el comportamiento con otros objetos
C. Object es la clase raíz de todos los demos objetos
D. Una subclase puede heredar de una sola superclase
E. Una subclase puede heredar de una o varias superases
F. Un paquete debe contener más de una clase

---

7. Given:

```
public class Triangle {
    static double area;
    int b = 2, h =3;
    public static void main(String[] args) {
        double p, b, h;// line n1
        if(area == 0){
            b = 3;
            h = 4;
            p = 0.5;
        }
        area = p * b * h; // line n2
        System.out.println("Area is " + area);
    }
}
```

What is the result?

A. Area is 6.0
B. Area is 3.0
C. Compilation fails at line n1.
D. Compilation fails at line n2.

8. Which two class definitions fail to compile?

```
A. final class A1{
    public A1(){}
}
B. public class A2 {
    private static int i;
    private A2(){}
}
C. final abstract class A5{
    protected static int i;
    void dostuff(){}
    abstract void doIt();
}
D. class A4{
    protected static final int i;
    private void doStuff(){}
}
E. abstract class A3{
    private static int i;
    public void doStuff(){}
    public A3(){}
}
```

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

9. Given:

```
      class C2 {
          public void displayC2(){
              System.out.print("c2");
          }
      }
      interface I{
          public void displayI();
      }
      class C1 extends C2 implements I{
          public void displayI(){
              System.out.print("c1");
          }
      }
```

And given the code fragment:

```
  C2 obj1 = new C1();
        I obj2 = new C1();
        C2 s = obj2;
        I t = obj1;
        t.displayI();
        s.displayC2();
```

What is the result?

A. C2C2
B. C1C2
C. C1C1
D. Compilation fails.

10. Given the code fragment:

```
public static void main(String[] args) {
    int[][] arr = new int[2][4];
    arr[0] = new int[]{1,3,5,7};
    arr[1] = new int[]{1,3};
    for(int[] a: arr){
        for(int i : a){
            System.out.print(i + " ");
        }
        System.out.println();
    }
}
```

What is the result?

A. 1 3 followed by an ArrayIndexOutOfBoundsException
B. 1 3 1 3 00
C. Compilation fails.
D. 1 3 13
E. 1 3 5 7 13