Que-1: Which constructors can be inserted at (1) in MySub without causing a compile-time error?

```
class MySuper {
        int number;
        MySuper(int i) {
                number = i;
                }
        }
class MySub extends MySuper {
        int count;
        MySub(int count, int num) {
                super(num);
                this.count = count;
        }
        // (1) INSERT CONSTRUCTOR HERE
}
```

Select the one correct answer.

(a) MySub() {}

(b) MySub(int count) { this.count = count; }

(c) MySub(int count) { super(); this.count = count; }

(d) MySub(int count) { this.count = count; super(count); }

(e) MySub(int count) { this(count, count); }

(f) MySub(int count) { super(count); this(count, 0); }

Que-2: What will the following program print when run?

```
// Filename: MyClass.java
public class MyClass {
        public static void main(String[] args) {
                B b = new B("Test");
                }
        }
`       class A {
                A() {
                        this("1", "2");
                }
                A(String s, String t) {
                this(s + t);
                }
                A(String s) {
                System.out.println(s);
                }
        }
        class B extends A {
                B(String s) {
                        System.out.println(s);
                }
                B(String s, String t) {
                        this(t + s + "3");
                }
                B() {
                        super("4");
                };
        }
```

Select the one correct answer.

(a) It will just print Test.

(b) It will print Test followed by Test.

(c) It will print 123 followed by Test.

(d) It will print 12 followed by a Test.

(e) It will print 4 followed by a Test.

Que-3: Which statements about interfaces are true?

(a) Interfaces allow multiple implementation inheritance.

(b) Interfaces can be extended by any number of interfaces.

(c) Interfaces can extend any number of interfaces.

(d) Members of an interface are never static.

(e) Members of an interface can always be declared static.

Que-4: Which statement is true about the following code?

```
// Filename: MyClass.java
abstract class MyClass implements Interface1, Interface2 {
        public void f() { }
        public void g() { }
}
interface Interface1 {
        int VAL_A = 1;
        int VAL_B = 2;
        void f();
        void g();
}
interface Interface2 {
        int VAL_B = 3;
        int VAL_C = 4;
        void g();
        void h();
}
```

Select the one correct answer.

(a) MyClass only implements Interface1. Implementation for void h() from Interface2 is missing.

(b) The declarations of void g() in the two interfaces conflict, therefore, the code will not compile.

(c) The declarations of int VAL_B in the two interfaces conflict, therefore, the code will not compile.

(d) Nothing is wrong with the code, it will compile without errors.

Que-5: What will be the result of compiling and running the following program?

```
public class Polymorphism2 {
        public static void main(String[] args) {
                A ref1 = new C();
                B ref2 = (B) ref1;
                System.out.println(ref2.g());
        }
}
class A {
        private int f() { return 0; }
        public int g() { return 3; }
}
class B extends A {
        private int f() { return 1; }
        public int g() { return f(); }
}
class C extends B {
        public int f() { return 2; }
}
```

Select the one correct answer.

(a) The program will fail to compile.

(b) The program will compile and print 0, when run.

(c) The program will compile and print 1, when run.

(d) The program will compile and print 2, when run.

(e) The program will compile and print 3, when run.

Que-6: Which of these field declarations are legal within the body of an interface? Select the three correct answers.

(a) public static int answer = 42;
(b) int answer;
(c) final static int answer = 42;
(d) public int answer = 42;
(e) private final static int answer = 42;

Que-7: Which declaration can be inserted at (1) without causing a compilation error? interface MyConstants

```
{
        int r = 42;
        int s = 69;
        // (1) INSERT CODE HERE
}
```

Select the two correct answers.

(a) final double circumference = 2 * Math.PI * r;

(b) int total = total + r + s;

(c) int AREA = r * s;

(d) public static MAIN = 15;

(e) protected int CODE = 31337;

Que-8: What will happen when you compile and run the following code?

```
public class Test{
    public static void main(String[] args){
        int i = 0, j = 3;
        for(;i < 3 && j > 0;i++, j--);
        {
            System.out.print(i + " " + j + ", ");
        }
    }
}
```

(a) 0 3, 1 2, 2 1,
(b) 1 2, 2 1,
(c) 0 3, 1 2, 2 1, 3 0,
(d) None of the above

Ans: System.out.print statement prints values as “3 0, “.

7. Given the following type and reference declarations, which assignment is legal?

```
// Type declarations:

interface I1 {}
interface I2 {}
class C1 implements I1 {}
class C2 implements I2 {}
class C3 extends C1 implements I2 {}

// Reference declarations:

C1 obj1;
C2 obj2;
C3 obj3;
```

Select the one correct answer.
(a) obj2 = obj1;
(b) obj3 = obj1;
(c) obj3 = obj2;
(d) I1 a = obj2;
(e) I1 b = obj3;
(f) I2 c = obj1;

11. What is the label of the first line that will cause compilation to fail in the following program?

```
 // Filename: MyClass.java
class MyClass {

        public static void main(String[] args) {

                MyClass a;
                MySubclass b;
                a = new MyClass();                      // (1)
                b = new MySubclass();                   // (2)
                a = b;                                  // (3)
                b = a;                                  // (4)
                a = new MySubclass();                   // (5)
                b = new MyClass();                      // (6)
        }
}
class MySubclass extends MyClass {}
```

4. Fill in the blanks in the following scope of access modifiers table:

· Write Y where access is possible and N where not.

Access Modifier	within Class	within package and outside Class	Outside Package and subclass	Outside package and not subclass
Private	e.g.- Y			
Default				
Protected				
Public				

Access Modifier	within class	within package	outside package by subclass only	outside package
Private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y

Q .10 What would be the result of compiling and running the following program?

```
class Vehicle {
        static public String getModelName() { return "Volvo"; }
        public long getRegNo() { return 12345; }
}
class Car extends Vehicle {
        static public String getModelName() { return "Toyota"; }
        public long getRegNo() { return 54321; }
}
public class TakeARide {
        public static void main(String args[]) {
                Car c = new Car();
                Vehicle v = c;
                System.out.println("|" + v.getModelName() + "|" + c.getModelName() +
"|" + v.getRegNo() + "|" + c.getRegNo() + "|");
        }
}
```

Select the correct answer.

(a) The code will fail to compile.

(b) The code will compile and print |Toyota|Volvo|12345|54321|, when run.

(c) The code will compile and print |Volvo|Toyota|12345|54321|, when run.

(d) The code will compile and print |Toyota|Toyota|12345|12345|, when run.

(e) The code will compile and print |Volvo|Volvo|12345|54321|, when run.

(f) The code will compile and print |Toyota|Toyota|12345|12345|, when run.

(g) The code will compile and print |Volvo|Toyota|54321|54321|, when run.

7.17 Which statements about the keywords extends and implements are true? Select the correct answers.

(a) The keyword extends is used to specify that an interface inherits from another interface.

(b) The keyword extends is used to specify that a class implements an interface.

(c) The keyword implements is used to specify that an interface inherits from another interface.

(d) The keyword implements is used to specify that a class inherits from an interface.

(e) The keyword implements is used to specify that a class inherits from another class.

7.11 What would be the result of compiling and running the following program?

```
final class Item {
        Integer size;
        Item(Integer size) { this.size = size; }
        public boolean equals(Item item2) {
        if (this == item2) return true;
        return this.size.equals(item2.size);
}
}
public class SkepticRide {
        public static void main(String[] args) {
                Item itemA = new Item(10);
                Item itemB = new Item(10);
                Object itemC = itemA;
                System.out.println("|" + itemA.equals(itemB) +
                "|" + itemC.equals(itemB) + "|");

        }
}
```

Select the one correct answer.

(a) The code will fail to compile.

(b) The code will compile and print |false|false|, when run.

(c) The code will compile and print |false|true|, when run.

(d) The code will compile and print |true|false|, when run.

(e) The code will compile and print |true|true|, when run.

Complete the following code snippet for desired output.

```
class Box {
      double width;
      double height;
      double depth;
      Box(double w, double h, double d) {
            _________;
            _________;
            _________;
      }
      public String toString() {
            return
      ____________________________________________________;
      }

}
```

```
class Demo {

      public static void main(String args[]) {
                  Box b = new Box(10, 12, 14);
                  String s  = _________________;// concatenate Box object
                  System.out.println(b);
                  System.out.println(s);
            }

}
```

Output:

Dimensions are 10.0 by 14.0 by 12.0

Box b: Dimensions are 10.0 by 14.0 by 12.0

Answers:

return "Dimensions are " + width + " by " + depth + " by " + height + ".";

String s = "Box b: " + b; // concatenate Box object