OOPs Assignment

Chapter 07-OOPs:

Q. Which statements are true?
Select the two correct answers.

1. In Java, the extends clause is used to specify the inheritance relationship.
2. The subclass of a non- abstract class can be declared abstract .
3. All members of the superclass are inherited by the subclass.
4. A final class can be abstract .
5. A class in which all the members are declared private , cannot be declared public .

Q. Which statements are true? Select the two correct answers.

1. A class can only be extended by one class.
2. Every Java object has a public method named equals .
3. Every Java object has a public method named length .
4. A class can extend any number of classes.
5. A non- final class can be extended by any number of classes.

Q. Which statements are true? Select the two correct answers.
(a) A subclass must define all the methods from the superclass.
(b) It is possible for a subclass to define a method with the same name and parameters as a method defined by the superclass.
(c) It is possible for a subclass to define a field with the same name as a field defined by the superclass.
(d) It is possible for two classes to be the superclass of each other.

Q.Given the following classes and declarations, which statements are true?
// Classes

```
class Foo {
        private int i;
        public void f() { /* ... */ }
        public void g() { /* ... */ }
}
class Bar extends Foo {
        public int j;
        public void g() { /* ... */ }
}
```

```
// Declarations:
        Foo a = new Foo();
        Bar b = new Bar();
```

Select the three correct answers.

1. The Bar class is a subclass of Foo .
2. The statement b.f(); is legal.
3. The statement a.j = 5; is legal.
4. The statement a.g(); is legal.
5. The statement b.i = 3; is legal.

Q. Which statement is true?
Select the one correct answer.
(a) Private methods cannot be overridden in subclasses.
(b) A subclass can override any method in a superclass.
(c) An overriding method can declare that it throws checked exceptions that are not thrown by the method it is overriding.
(d) The parameter list of an overriding method can be a subset of the parameter list of the method that it is overriding.
(e) The overriding method must have the same return type as the overridden method.

Q. Given classes A , B , and C , where B extends A , and C extends B , and where all classes implement the instance method void doIt() . How can the doIt() method in A be called from an instance method in C ? Select the one correct answer.

(a) doIt();
(b) super.doIt();
(c) super.super.doIt();
(d) this.super.doIt();
(e) A.this.doIt();
(f) ((A) this).doIt();
(g) It is not possible.

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

```
// Filename: MyClass.java
public class MyClass {
  public static void main(String[] args) {
    C c = new C();
    System.out.println(c.max(13, 29));
  }
}

class A {
  int max(int x, int y) { if (x>y) return x; else return y; }
}

class B extends A{
  int max(int x, int y) { return super.max(y, x) - 10; }
}

class C extends B {
  int max(int x, int y) { return super.max(x+10, y+10); }
}
```

Select the one correct answer.
(a) The code will fail to compile because the max() method in B passes the argu-
ments in the call super.max(y, x) in the wrong order.
(b) The code will fail to compile because a call to a max() method is ambiguous.
(c) The code will compile and print 13 , when run.
(d) The code will compile and print 23 , when run.
(e) The code will compile and print 29 , when run.
(f) The code will compile and print 39 , when run.

Q. Which is the simplest expression that can be inserted at (1), so that the program prints the value of the text field from the Message class?

```
// Filename: MyClass.java
class Message {
  // The message that should be printed:
  String text = "Hello, world!";
}

class MySuperclass {
  Message msg = new Message();
}

public class MyClass extends MySuperclass {
  public static void main(String[] args) {
    MyClass object = new MyClass();
    object.print();
  }

  public void print() {
    System.out.println( /* (1) INSERT THE SIMPLEST EXPRESSION HERE */ );
  }
}
```

Select the one correct answer.
(a) text
(b) Message.text
(c) msg.text
(d) object.msg.text
(e) super.msg.text
(f) object.super.msg.text

Q. Which method declarations, when inserted at (7), will not result in a compile-time error?

```
class MySuperclass {
  public        Integer step1(int i)                    { return 1; }     // (1)
  protected     String  step2(String str1, String str2) { return str1; }  // (2)
  public        String  step2(String str1)              { return str1; }  // (3)
  public static String  step2()                         { return "Hi"; }  // (4)

  public MyClass      makeIt()  { return new MyClass(); }                 // (5)
  public MySuperclass makeIt2() { return new MyClass(); }                 // (6)
}

public class MyClass extends MySuperclass {
  // (7) INSERT METHOD DECLARATION HERE
}
```

Select the two correct answers.
(a) public int step1(int i) { return 1; }

(b) public String step2(String str2, String str1) { return str1; }
(c) private void step2() { }
(d)private static void step2() { }
(e)private static String step2(String str) { return str; }
(f)public MySuperclass makeIt() { return new MySuperclass(); }
(g)public MyClass makeIt2() { return new MyClass(); }

Q. 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 one 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.

Q. 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.

1. The code will fail to compile.
2. The code will compile and print |false|false| , when run.
3. The code will compile and print |false|true| , when run.
4. The code will compile and print |true|false| , when run.
5. The code will compile and print |true|true| , when run.

Q. 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); }

Q .Which statement is true? Select the one correct answer.
(a) A super() or this() call must always be provided explicitly as the first statement in the body of a constructor.
(b) If both a subclass and its superclass do not have any declared constructors, the implicit default constructor of the subclass will call super() when run.
(c) If neither super() nor this() is declared as the first statement in the body of a constructor, this() will implicitly be inserted as the first statement.
(d) If super() is the first statement in the body of a constructor, this() can be declared as the second statement.
(e) Calling super() as the first statement in the body of a constructor of a subclass will always work, since all superclasses have a default constructor.

Q. 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.

1. It will just print Test .

2. It will print Test followed by Test .
3. It will print 123 followed by Test .
4. It will print 12 followed by Test .
5. It will print 4 followed by Test .

Q. Which statements about interfaces are true?
Select the two correct answers.

1. Interfaces allow multiple implementation inheritance.
2. Interfaces can be extended by any number of interfaces.
3. Interfaces can extend any number of interfaces.
4. Members of an interface are never static.
5. Members of an interface can always be declared static .

Q. 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;

Q. Which statements about the keywords extends and implements are true?
Select the two 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.

Q. Which statement is true about the following code?

```java
// 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.

Q. Which declaration can be inserted at (1) without causing a compilation error?

```java
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;

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

```
public class Polymorphism {
  public static void main(String[] args) {
    A ref1 = new C();
    B ref2 = (B) ref1;
    System.out.println(ref2.f());
  }
}

class A           { int f() { return 0; } }
class B extends A { int f() { return 1; } }
class C extends B { int f() { return 2; } }
```

Select the one correct answer.

(a) The program will fail to compile.
(b) The program will compile but will throw a `ClassCastException`, when run.
(c) The program will compile and print `0`, when run.
(d) The program will compile and print `1`, when run.
(e) The program will compile and print `2`, when run.

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

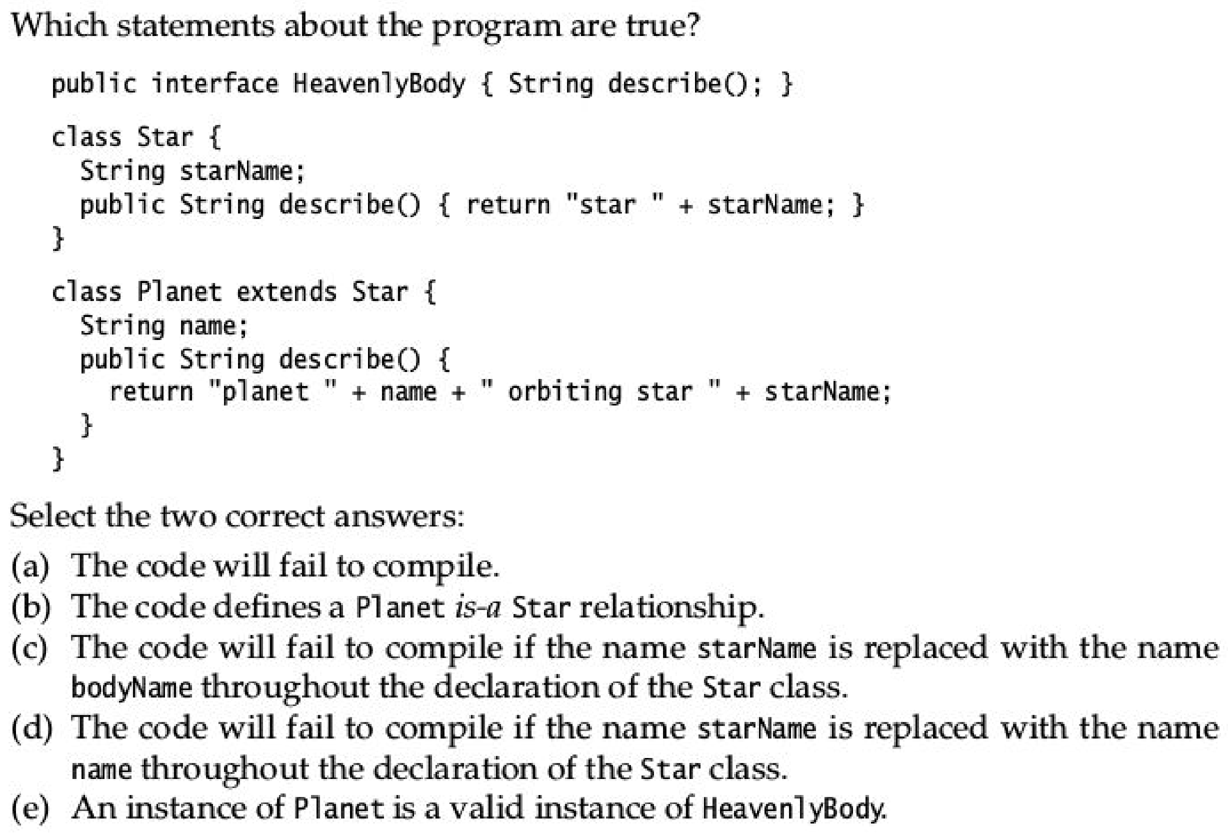
```
public class Polymorphism2 {
  public static void main(String[] args) {
    A ref1 = new C();
    B ref2 = (B) ref1;
    System.out.println(ref2.g());
  }
}

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.

1. The program will fail to compile.
2. The program will compile and print 0 , when run.
3. The program will compile and print 1 , when run.
4. The program will compile and print 2 , when run.

5. The program will compile and print 3 , when run.

Q.



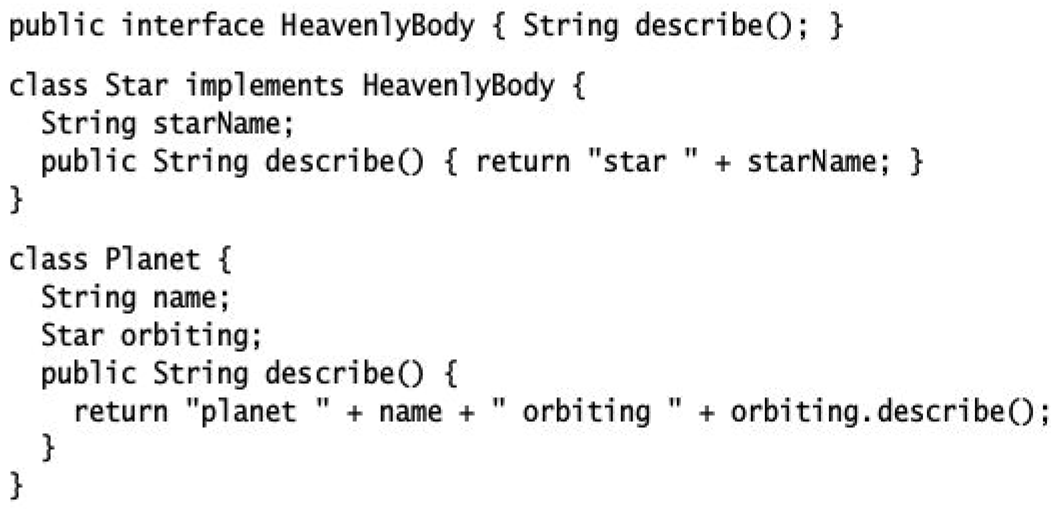
Which statements about the program are true?

```
public interface HeavenlyBody { String describe(); }

class Star {
  String starName;
  public String describe() { return "star " + starName; }
}

class Planet extends Star {
  String name;
  public String describe() {
    return "planet " + name + " orbiting star " + starName;
  }
}
```

Select the two correct answers:

(a) The code will fail to compile.
(b) The code defines a `Planet` *is-a* `Star` relationship.
(c) The code will fail to compile if the name `starName` is replaced with the name `bodyName` throughout the declaration of the `Star` class.
(d) The code will fail to compile if the name `starName` is replaced with the name `name` throughout the declaration of the `Star` class.
(e) An instance of `Planet` is a valid instance of `HeavenlyBody`.

Q. Given the following code, which statement is true?



```
public interface HeavenlyBody { String describe(); }

class Star implements HeavenlyBody {
  String starName;
  public String describe() { return "star " + starName; }
}

class Planet {
  String name;
  Star orbiting;
  public String describe() {
    return "planet " + name + " orbiting " + orbiting.describe();
  }
}
```

Select the one correct answer:
(a) The code will fail to compile.

(b) The code defines a Planet has-a Star relationship.
(c) The code will fail to compile if the name starName is replaced with the name bodyName throughout the declaration of the Star class.
(d) The code will fail to compile if the name starName is replaced with the name name throughout the declaration of the Star class.
(e) An instance of Planet is a valid instance of a HeavenlyBody .

Q. Which statement is not true?
Select the one correct answer.
(a) Maximizing cohesion and minimizing coupling are the hallmarks of a well-designed application.
(b) Coupling is an inherent property of any non-trivial OO design.
(c) Adhering to the JavaBeans naming standard can aid in achieving encapsulation.
(d) Dependencies between classes can be minimized by hiding implementation details.
(e) Each method implementing a single task will result in a class that has high cohesion.
(f) None of the above.