Cognizant Technology Solutions

Inheritance,Abstract Classes and Interface Exercise

CATP Java Team
5/20/2012

For The Associates:

The documents details two flavors of problem statements

- **Statement # 1:** Few problem solutions have been provided for associates should analyze the program and write down the program output. This will enhance the analyzing skills of associates and also understand "why" part of java programming feature. The associates can then try running the program in eclipse and check if the output with what they have written.
- **Stamen # 2:** There are some problem statements provided similar to the final assessment and associates need to solve it. This will enhance the programming skills of the associates.
  IMPORTANT: These exercises will gear you up for the core java assessment so please develop/analyze the exercise independently. In case you are stuck up reach out to the trainers.

Exercises:

1. What is the expected output?

```
class Plant {
Plant() {
System.out.println("Plant created");
}
}
class Tree extends Plant {
Tree() {
System.out.println("Tree created");
super();
}
}
public class Test {
public static void main(String args[]) {
Tree tree = new Tree();
}
}
```

2. What is the result of compiling and running the following code?

```
class Base {
private Base() {System.out.print("Base");}
}
public class Derived extends Base {
public Derived() {System.out.print("Derived");}
public static void main(String[] args) {
new Derived();
}
}
```

3. Can the following interface declaration compile?

```
abstract interface Bendable {
final int x = 2009;
void method1() ;
public static class Angle {}
}
```

4. Fix compilation errors if in the following code snippet

```
abstract class AirPlane {
abstract void fly();
void land() {
System.out.print("Landing..");
}
}
class AirJet extends AirPlane {
AirJet() {
super();
}
void fly() {
System.out.print("Flying..");
}
abstract void land() ;
}
```

5. What is the expected output?

```
interface Foldable {
public void fold() throws Exception ;
}
class Paper implements Foldable {
public void fold() {
System.out.print("Fold");
}
}
public class Tester {
public static void main(String args []) {
Foldable obj1 = new Paper();
obj1.fold();
Paper obj2 = new Paper();
obj2.fold();
}
}
```

6. Will the following code compile?

```
class Creature {
void grow() {
}
}
class Bird extends Creature {
void fly() {
}
}
class Falcon extends Bird {
void hunt() {
}
}
public class Tester {
public static void main(String[] args) {
Creature c1 = new Bird();
Falcon c2 = new Falcon();
Falcon c3=new Bird();
Falcon c4=new Creature();
Bird c5=new Falcon();
Bird c6=new Creature();
}
}
```

7. What is the result of compiling and running the following code?

```
class Creature {
String getName() {
return "Creature";
}
}
class Bird extends Creature {
String getName() {
return "Bird";
}
boolean hasFeather(){
return true;
}
}
public class Tester {
public static void main(String[] args) {
Creature c1=new Bird();
System.out.println(c1.getName());
System.out.println(c1.hasFeather());
Creature c2=new Creature();
System.out.println(c2.getName());
}
}
```

8. What is the output?

```
class Employee{
String empId;
Employee(String empId){
 this.empId=empId;
System.out.println("Name : "+ this.empId);
}
}
class Manager extends Employee{
int salary;
Manager(String name, int salary){
this.salary=salary;
System.out.println("Salary : "+ this.salary);
}
}
public static void main(String [] args){
Manager manager=new Manager();
}
}
```

9. Will the following code compile

```
interface Colorable {}
class Vehicle {}
class Car extends Vehicle implements Colorable {}
public class Tester {
public static void main(String[] args) {
Vehicle a = new Car();
Colorable i = (Colorable) a;
Vehicle b = new Vehicle();
Colorable j = (Colorable) b;
}
}
```

10. What is the result of compiling and running the following code?

```
interface Colorable { }
interface Bouncable extends Colorable { }
class Super implements Bouncable { }
class Sub extends Super implements Bouncable { }
class Individual { }
public class Tester {
public static void main(String[] args) {
System.out.print(new Sub() instanceof Super);
System.out.print(new Sub() instanceof Colorable);
System.out.print(new Super() instanceof Sub);
System.out.print(new Individual() instanceof Super);
}
}
```

11. What is the result of compiling and running the following code?

```
public class Tester {
int x = 12;
static Tester reset( Tester obj) {
obj = null;
return obj;
}
public static void main(String[] args) {
Tester o1 = new Tester();
o1 = reset(o1);
System.out.print(o1.x);
}
}
```

12. What is the result of compiling and running the following code?

```
public class A {
private void printName(){
System.out.println(“Value-A”);
}
}
public class B extends A{
public void printName(){
System.out.println(“Name-B”);
}
}
public class Test{
public static void main (String[] args) {
B b = new B();
b.printName();
}
}
```