

Quiz W3 – CS 102

Name: _____, ID: _____

1) Consider the following code:

```
import java.util.ArrayList;

interface Observer
{
    void observe();
}

abstract class Person
{
    private String name;
    public Person(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
    public abstract void look();
    public abstract void think();
}

class Student extends Person implements Observer
{
    private int id;
    public Student(String name, int id) {
        super(name);
        this.id = id;
    }
    public int getId() {
        return id;
    }
    public void look() {
        System.out.print("Looking...");
    }
    public void think() {
        System.out.print("Thinking...");
    }
    public void observe() {
        look(); think();
        System.out.println("Observed.");
    }
}

class Robot implements Observer
{
    private String model;
    private String id;
    public Robot(String model, String id) {
        this.model = model;
        this.id = id;
    }
    public String getModel() {
        return model;
    }
    public String getId() {
        return id;
    }
    public void sense() {
        System.out.print("Sensing...");
    }
    public void analyze() {
        System.out.print("Analyzing...");
    }
    public void observe() {
        sense(); analyze();
        System.out.println("Observed.");
    }
}

public class Quiz2
{
    public static void main(String[] args) {
        ArrayList<Observer> a = new ArrayList<Observer>();
        a.add(new Student("Andy", 12345));
        a.add(new Robot("C-3PO", "as23478"));
        Observer o0 = a.get(0);
        Observer o1 = a.get(1);
        o0.observe();
        o1.observe();
        o0 = o1;
        o0.observe();
        (a.get(0)).observe();
    }
}
```

a) Print the program output below.

b) Write a new class `Instructor` that extends from the `Person` class and implements the `Observer` interface. The `Instructor` class must have a `staffId` instance variable of type `int` and a `department` instance variable of type `String`. Write an appropriate constructor to initialize these fields. Make sure the class compiles without issues.

2) Consider the following code:

```
abstract class Person {
    public abstract void greet();
}
class Student extends Person {
    public void greet() {
        System.out.println("No pain, no gain.");
    }
}
class Instructor extends Person {
    public void greet() {
        System.out.println("Hunger is the best sauce.");
    }
}
class Professor extends Instructor {
    public void greet() {
        super.greet();
        System.out.println("Rome wasn't built in a day.");
    }
}
```

a) What will be the output of the following program?

```
Person[] people = { new Student(), new Instructor(), new Professor() };
for (Person pl: people) {
    pl.greet();
}
```

b) Write the type of the conversion (one of *narrowing*, *widening*, *invalid*) for each and whether the code compiles and runs:

```
Person p = new Student();           // conversion: _____, compiles: ____, runs: ____
Instructor i = new Instructor();    // conversion: _____, compiles: ____, runs: ____
Student s = (Student) i;           // conversion: _____, compiles: ____, runs: ____
Professor r = new Professor();      // conversion: _____, compiles: ____, runs: ____
p = r;                              // conversion: _____, compiles: ____, runs: ____
i = (Instructor) p;                // conversion: _____, compiles: ____, runs: ____
r = (Professor) i;                 // conversion: _____, compiles: ____, runs: ____
Student s1 = (Student) p;          // conversion: _____, compiles: ____, runs: ____
```