

QUEENS COLLEGE  
CSCI 212

Department of Computer Science  
Final Exam Summer 2017 08.15.17

Instructor: Krishna Mahavadi

03.00pm – 04.00pm Tuesday, August 15, 2017

Complete all of the following information.

**STUDENT LAST NAME** (PRINT): \_\_\_\_\_

**STUDENT First NAME** (PRINT): \_\_\_\_\_

**Cuny 1st ID#**: \_\_\_\_\_

Lab Instructor: \_\_\_\_\_

**THIS IS A CLOSED BOOK TEST. NO BOOKS, NOTES, COMPUTERS, CELL PHONES, OR CALCULATORS ARE ALLOWED.**

**It is department policy to give a grade of F to any student who helps or receives help from any other student during an exam.**

**The exam has 5 problems, you should answer all of them.**

**If your printed exam is missing any problem please notify the proctor as soon as possible.**

**Answer the problems in the spaces provided.**

problem:	01	02	03	04	05	$\Sigma$
grade:						

**Problem 1** (12 points) In this problem you will create an *interface DiscountPolicy* and an implementing *class BulkDiscount*. These are to be used to calculate the discount percentage to be applied on the purchase of some number of a single item. The discount percentage has type *double* and the number of items has type *int*.

(a): Create an *interface DiscountPolicy*. It should define one method *computeDiscount* that will return the discount for the purchase of a given number of a single item. The method has two parameters, *count* and *itemCost*.

**Answer:**

(b): Write an implementing *class BulkDiscount*. It should have a constructor with two parameters that represent the minimum quantity of items that would qualify for a discount and the percentage discount given to qualifying purchases. Implement the method *computeDiscount* so that if the number of items purchased is more than the minimum, then the discount percentage is returned, otherwise there is no discount and zero is returned.

**Answer:**

**Problem 2** (6 points) Each of the following 3 pieces of code might contain an error. For each part, if there is an error correct it by rewriting corrected or additional code in the space provided. If the code has no error, write **No Error** in the space.

(a):

```
// Copy an array
public static int[] copyArray(int[] anArray) {
    int[] temp = new int[anArray.length];
    temp = anArray;
    return temp;
}
```

Answer:

(b):

```
public void doStuff(int n){
    if (n < 0)
        throw new Exception("Negative number.");
}
```

Answer:

(c): Read the following *class Fruit* and the *class FruitDemo*. The program will compile and run but produce the following run-time error.

Exception in thread "main" java.lang.ClassCastException: Fruit cannot be cast to java.lang.Comparable.

```
import java.util.Arrays;

public class Fruit{
    private String fruitName;
    public Fruit() {fruitName = "";}
    public Fruit(String name) {
        fruitName = name;}
    public String getFruitName(){
        return fruitName;}
}

public class FruitDemo {
    public static void main(String[] args) {
        Fruit[] fruits = new Fruit[3];
        fruits[0] = new Fruit("Pineapple");
        fruits[1] = new Fruit("Durian");
        fruits[2] = new Fruit("Banana");
        Arrays.sort(fruits);
        for(Fruit f: fruits)
            System.out.println(f.getName());
    } }
```

Answer:

**Problem 3** (12 points) Assume that a *class Vehicle* has instance variables *passengers*, *fuelCapacity* and *mpg* (miles per gallon which is an indicator of how far a Vehicle can travel on 1 gallon of gasoline).

Write a derived *class Truck* that extends *class Vehicle*. A *Truck* has just one additional instance variable *cargoCapacity*.

(a): Write the derived class default constructor, followed by a constructor with the three parameters for the *class Vehicle* and also the additional parameter for the derived class. **Note: Do not write vehicle class constructor or methods.**

**Answer:**

(b): Write an accessor and setter method for the instance variable *cargoCapacity* of the truck class.

**Answer:**

(c): Write a method to find the number of gallons of fuel needed to travel  $x$  miles. This method needs a double parameter  $x$  and returns a double.

**Answer:**

(d): Override the *Vehicle* class's *toString* method by writing a *toString* method for the derived class. It should return an answer obtained by using the *Vehicle* *toString* method and adding on the value of *cargoCapacity*.

**Answer:**

**Problem 4** (8 points) What will be printed by this program?

```
public class Question4 {
    public static void main (String[] args){

        B b = new B();
        A a = b;

        System.out.println(a);           // line A
        b.increment();
        System.out.println(a);           // line B
        a = new A();
        System.out.println(a.getB());    // line C
        System.out.println(b.getA());    // line D
    }
}
```

The two classes used are implemented as follows.

```
class A {
    int x = 0;

    public void increment() {
        x++;
    }

    public String toString() {
        return "class A: " + x;
    }

    public B getB() {
        return new B( );
    }
}
```

```
class B extends A{
    public String toString( ) {
        return "Class B: " + x;
    }

    public A getA( ) {
        return getB( );
    }
}
```

(a) Output generated at line A is:

(b) Output generated at line B is:

(c) Output generated at line C is:

(d) Output generated at line D is:

**Problem 5** (12 points) Write static methods for a *class P5* that carry out the indicated tasks.

(a) Write a method to find the the number *false* elements in an *ArrayList* of of *Boolean* values.

**Answer:**

(b) Write a method that creates a *JFrame* object and add to its window three *Jbutton* objects and three *JTextField* objects next to them,

**Answer:**

(c) Messages in Morse code are composed of two signals: dot and dash. Dot takes 2 microseconds to send, and dash takes 3 microseconds to send. Let  $m(k)$  be the number of different Morse code messages that take exactly  $k$  microseconds to send.

If  $k$  is 2 or 3,  $m(k)$  is 1 message (dot or dash respectively). If  $k$  is larger than 3,  $m(k) = m(k-2) + m(k-3)$  (since the two terms count message beginning with a dor and a dash).

Write a recursive static method that returns the the value of  $m(k)$ . The value of  $k$  is an argument in the method call.

**Answer:**