QUEENS COLLEGE Department of Computer Science
CSCI 212 Second Midterm Exam Spring 2018 04.30.18

Instructor: Krishna Mahavadi
08.00am – 08.50am, Monday, April 30, 2018

1. Read the following code. (*10 pts*)

```
class A {
   void method(char ch) {  System.out.println("A.method() " + ch);i }
   String what() { return "Returning A";}
   void adjust(){ System.out.println("Adjusting A");}
}

class B extends A {
   void method(char ch) { System.out.println("B.method() " + ch);}
   String what() {return "Returning B";}
   void adjust() {System.out.println("Adjusting B");}
}

class C extends A {
   void method(char ch) {System.out.println("C.method() " + ch);}
   String what() {return "Returning C";}
   void adjust() {System.out.println("Adjusting C");}
}

class D extends A {
   void method(char ch) { System.out.println("D.method() " + ch);}
   String what() {return "returning D";}
   void change() {System.out.println("Changed String"); }
   void adjust() {System.out.println("Adjusting D"); }
}

class E extends B {
   void method(char ch) { System.out.println("E.method() " + ch); }
   String what() { return "Returning F"; }
}

class F extends B {
   void method(char ch) { System.out.println("F.method() " + ch); }
   void adjust() { System.out.println("Adjusting F"); }
}

public class Q1 {
    public static void main(String[] args) {
       A a = new C();
       System.out.println(a.what()); //-----(a)
       A a1 = new E();
       a1.method('X');//-----(b)
       B b = new B();
       b.adjust(); //--------(c)
       B b1 = new F();
```

```
        System.out.println(b1.what()); //----(d)
        A a2 = new D();
        ((D)a2).change(); //-------(e)
    }
}
```

(a) What is printed on line (a)?
Answer: Returning C

(b) What is printed at line (b)?
Answer: E.method() X

(c) What is printed at line (c)?
Answer: Adjusting B

(d) What is printed at line (d)?
Answer: Returning B

(e) What is printed at line (e)?
Answer: Changed String

2. For this question you need to write some methods and class headers.

(a) Assume that you have written a Rectangle class with instance variables length and width. You have already written all set and get methods and area and perimeter methods. **Write an equals() method** that takes Object o as a parameter. The method should return true when the Object o is a rectangle with the same length and width. (*2pts*)
Answer:

```
public boolean equals(Object o){
   boolean isEqual = false;
   if(o != null && o instanceof Rectangle) {
      Rectangle other = (Rectangle)o;
      isEqual = (this.length == other.length &&
              this.width == other.width);
   }
   return isEqual;
}
```

(b) A class named Fruit implements an interface called Edible. The interface has a single method called howToEat(). A class called Orange extends Fruit and implements Edible. **Write the class header for the Orange class and override the howToEat() method** of the Fruit class. The method should print a brief message to the screen about how to eat an orange. Do not write any other methods or constructors. (*2pts*)
Answer:

```
public class Oragne extends Fruit implements Edible {
   @Override
   pubic void howToEat() {
      System.out.println("Peel it and eat, or juice it and drink");
   }
}
```

(c) Write an inner class called Destination inside the class Parcel. This class has one instance variable called label of type String. It has one constructor that has a parameter of type String to initialize the instance variable. It has a single method called readLabel that just returns the label. **Write the class headers for Parcel and Destination and the constructor and readLabel methods for the class Destination.** Do not write any methods of the Parcel class. (*2pts*)
Answer:

```
public class Parcel{
   class Destination {
      private String label;
```

```
      Destination (String l){
         label = l;
      }

      private string readLabel() {
         return label;
      }
   }
}
```

(d) Use a PrintWriter to write data to a file called myData.txt. The data should be 20 numbers generated by Math.random() * 10. Write one number per line. I do not want you to write the entire class. Assume that you have imported the required classes etc. Your code should open the file and write data to it. Use a try catch block to catch any Exceptions. **Write only a main method that performs these tasks.** (*4pts*)

Answer:

```
import java.io.PrintWriter;
import java.io.File;
import java.io.FileNotFoundException;

public class TestWriter {
  public static void main(String[] args) {
     File fname = new File("data.text");
     PrintWriter writer = null;
     try{
        writer = new PrintWriter(fname);
        for (int i = 0; i < 20; i++)
            writer.println(((i+1) + " " + Math.random()*10);
      }
      catch(FileNotFoundException e){
         System.out.println("Error opening the file " + fname);
         System.exit(0);
      }
      writer.close();
   }
}
```

3. This question has two parts. Part 1 defines an interface and part 2 implements the interface in a class.

(a) **Create an interface called MessageEncoder that has a single abstract method encode(String plainText)**, where plainText is a message to be encoded. The method will return an encoded message as a String. (*2pts*)

Answer:

```
public interface MessageEncoder / //If you don't specify public, it only has package access
   public String encode(String plainText); //All interface methods must be public
/
```

(b) **Create a class called ShuffleCipher that implements the interface MessageEncoder that you wrote in part 1. The constructor should have one integer parameter called n. Define the method encode so that the message is shuffled n times.** To shuffle, split the message in half and then take the characters from each half alternately. For example, if the message is abcdefghi, the halves are abcde and fghi. The shuffled message is afbgchdie.
(*Hint:* You might want to define a private method that performs a single shuffle.) (*8 pts*)

```
public class ShuffleCipher implements MessageEncoder {
```

```
   private int numShuffles;
   public ShuffleCipher(int n) {
      numShuffles = n;
   }

   public String encode(String plainText) {
      String text = new String(plainText); //made a copy of plainText
      text = text.toLowerCase().replaceAll(" ",""); //Removed any spaces in the string
      for(int i = 0; i < numShuffles; i++)
          text = singleShuffle(text);
      return text;
   }

   private string singleShuffle(String str) {
      String shuffled = "";
      int len = (str.length() + 1)/2;
      String s1 = str(0, len);
      String s2 = str(len);
      if(s2.length() <= s1.length()) {
         for(int i = 0; i < s2.length(); i++){
            shuffled += s1.substring(i, i+1) +
                        s2.substring(i, i+1);
         }
      }
      if (len > s2.length())
         shuffled += s1.substring(len-1);
      return shuffled;
   }
}
```