

CMPE 261 – Mid-term Exam – 2013-2014 Fall

Student Full Name: _____ Student Number: _____

In all of the questions, give the answer in the provided space below or within each question. Do not write any answers on a separate paper!

PART A (54 points) Questions 1-7 on Code reading

1. (6 points) Consider the following definitions in a Java program:

```
int x = 1;
int y = x;
y += 1;
String s = "abc";
String t = s;
t += "def";
```

What would be the output when each of the statements below are executed after the above definitions? A statement may produce a value, or an error message. An example is given for the first statement:

	Statement	Output
	System.out.println(x)	1
1	System.out.println(y)	
2	System.out.println(z)	
3	System.out.println(s)	

2. (8 points) Consider the following definitions in a Java program:

```
int[] x = {1, 2, 3};
int[] y = x;
y[2] += 1;
String[] s = {"abc", "def"};
String[] t = s;
t[0] += "hgi";
```

What would be the output when each of the statements below are executed after the above definitions? A statement may produce a value, or an error message.

	Statement	Output
1	System.out.println(x[2])	
2	System.out.println(y[3])	
3	System.out.println(y[1])	
4	System.out.println(s[0])	

3. (8 points) Consider the following class definition in a Java program:

```
class C {  
    double x;  
    C (double x) { this.x = x; }  
}
```

and assume that the following statements are executed

```
C c1 = new C(1);  
C c2 = new C(2);  
C c3 = new C(3);  
c3 = c1;  
c1.x += 1;  
c3.x += 1;
```

answer the following questions:

(a) How many objects will there be in the memory after the above statements are executed?

(b) What will be the output produced when the following statements are executed after those above:

	Statement	Output
1	<code>System.out.println(c1.x)</code>	
2	<code>System.out.println(c2.x)</code>	
3	<code>System.out.println(c3.x)</code>	

4. (6 points) Consider the following class definition in a Java program:

```
class C {  
    String s;  
    static String f="f";  
    C (String s) { this.s = s; }  
}
```

and assume that the following statements are executed

```
C c1 = new C("a");  
C c2 = new C("b");  
C c3 = new C("c");  
c1 = c3;  
c1.f += "d";
```

What will be the output produced when the following statements are executed after those above:

	Statement	Output
1	<code>System.out.println(c1.s)</code>	
2	<code>System.out.println(c2.f)</code>	
3	<code>System.out.println(c3.s)</code>	

5. (10 points) In each of the questions below, check if the given code is correct (in the sense that it will compile without errors) and indicate so, or find and explain the error in the code:

	Code	Indicate if correct, or explain the error
1	<pre>interface I { } class C implements I { } class D { C x= new C(); I y=new C(); }</pre>	
2	<pre>Int[] x=3;</pre>	
3	<pre>interface I { } class C { I i=new I(); int j=1; }</pre>	
4	<pre>interface I { public int f(); } class C implements I { } class D { I x=new C(); }</pre>	
5	<pre>Object o = 3;</pre>	

6. (8 points) Consider the definitions below:

```
interface I { }
interface J { }
interface JJ extends J { }
class A { }
class B extends A { }
class C extends A implements J { }
class D implements JJ { }
class E extends B implements J { }
```

For each of the classes or types given on the left hand side in the table below, give a list of types with which it is compatible with. An example is given on the first line:

	Class	Type compatibility
	A	Object, A
1	B	
2	C	
3	D	
4	E	

7. (8 points) The following *incomplete* Java code implements a class to store an array of integers. It contains a method `max` which is intended to find and return the maximum value in the stored array, but the method is incomplete. Complete the method so that it will find and return the required value.

```
class IntegerArray {
    int[] data;
    IntegerArray(int[] data) {
        this.data=data;
    }

    /** Finds and returns the maximum value in the array */
    int max() {

    }
}
```

PART B (55 points) Questions 8-11 on Code design

- 8. (10 points)** Consider a problem where a car is identified with its model and make (e.g. “Honda” and “Civic”), year of production (e.g. 2012), and its price. Write a class definition named `Car` to represent car objects with these information. Make sure your class has a proper constructor.
- 9. (15 points)** Consider a study room in a university or in an office, which is identified with its room number (e.g. “S-E2-203”), and keeps track of room's occupied status. Everyone can “use” this study room if it is not occupied, and “leave” the room when they are finished using. In this problem you are asked to represent a study room object which provides the methods to use or leave the room, keeps track of its occupied status, and provides a method called “isOccupied” to learn its status. The “use” method must not succeed if the room is already occupied.

10. (15 Points) Consider an entertainment activity such as a movie session, a concert, a theatre play, or any planned event like those which has a certain capacity. If someone wants to buy tickets for such an event, they would want to check if there are enough seats, then proceed to buy tickets. The following interface captures these operations:

```
interface Activity {
    /** Returns the number of available seats */
    int remainingSeats();

    /** returns true if there are enough seats for given
     * many people */
    boolean haveEnoughSeats(int howManyPeople);

    /** Buy tickets for given number of people. Throws exception
     * if there are not enough seats left*/
    void buyTickets(int howManyPeople) throws Exception;
}
```

Write a class named `MovieSession` which represents a movie, and implements the above interface, thus keeps track of the number of available seats. Please note that this is a simplistic problem in which we do not keep track of who buys which seat, or we are not concerned with details such as payments. Keep your implementation simple, and limited to the stated problem.

11. (15 points) In this problem we define a pairing as the process of pairing a single object with a pair of objects of the same kind. An object may be paired, or may be alone/unpaired. The following interface provides methods that aim to capture the object's pairing status, as well as to obtain the object and its pair object:

```
interface Pairing<T> {
    /** returns the object */
    T getObject();

    /** returns whether object is paired */
    boolean isPaired();

    /** returns the pair if object is paired, throws
     * an exception if not */
    Pairing<T> getPair() throws Exception;
}
```

Note that one can construct a `list` similar to those in Racket programming, using this interface. For example, if we denote objects as O1, O2, O3, and pairing operation with parenthesis, a list of three objects would be a series of pairings (O3 (O2 (O1 null))).

(a) (10 points) Write a class named `PairingImplementation` which implements the above interface

(b) (5 points) Write statements to create two instances of your class such that the generic type is bound to `Integer`, and the first pairing (which is unpaired) is used as the pair of the second pairing similar to the example above.