Ryerson University
Department of Electrical & Computer Engineering
COE318/BME506

Final Examination Dec. 14, 2012

Name:_____ Section:_____

Student Number: _____

Time: 3 hours
Circle the name of your Professor: E. Bagheri, K. Clowes, O. Das, T. Yang.

Instructions

This exam is 3 hours long. It contains 10 questions. Please check your copy to make sure that it contains 18 pages (including this page) before you start.

(1) This is a closed-book exam.
(2) If in doubt on any question, then you must clearly state your own assumption(s).
(3) For each question, you must show your steps and reasoning clearly to be awarded part marks for your answer.

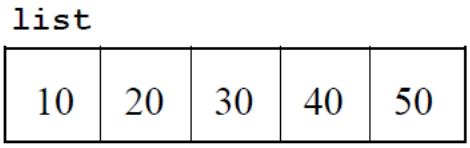
For marking use only (do not fill)

Questions	Score	Max
1		15
2		10
3		20
4		20
5		15
6		10
7		15
8		15
9		20
10		10
Total		150

Question 1. (15 marks)
Suppose that the integer array list has been declared and initialized as follows:

```
int[] list = { 10, 20, 30, 40, 50 };
```

This statement sets up an array of five elements with the initial values shown below:

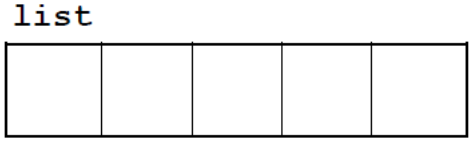


Given this array, what is the effect of calling `magicMethod(list);`

if `magicMethod` is defined as:

```
public void magicMethod(int[] array) {

      int tmp = array[array.length - 1];
     for (int i = array.length-1; i > 1; i--) {
          array[i-1] = array[i];
      }
      array[0] = tmp;

}
```

Work through the method carefully and indicate your answer by filling in the boxes below to show the final contents of list:



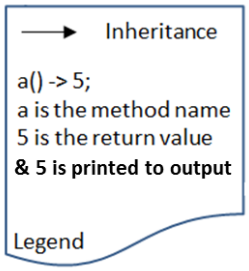
Answer: list
[50, 50, 50, 50, 50]

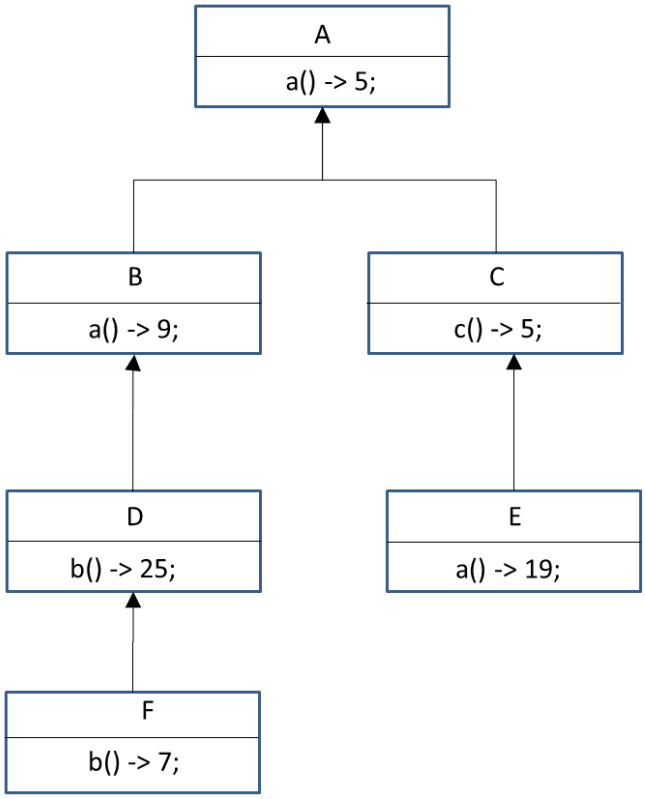
Question 2. (10 marks)
You are working for an online retail store that sells clothing. Your manager has asked you to modify the Customer class so that it assigns a new unique customer ID number when a customer object is created. The current implementation constructs a new Customer object without a customer ID..

```
/** A class representing a Customer */
public class Customer {

      // instance variables
      private int idNumber; // customer ID number
      private String name; // customer name
      private ShoppingCart items; // customer's shopping cart of
      // to-be-purchased items
      private Address mailingAddress; // customer's mailing address


      private static int count = 0;



      /** Construct a new customer object with name, and mailing address */
      public Customer(String name, Address mailingAddress) {

            count = count + 1;
            this.idNumber = count;


            this.name = name;
            this.mailingAddress = mailingAddress;
            this.items = new ShoppingCart();
      }
      // other methods omitted to save space
}
```

Make the required changes to the above code to have a unique customer ID be given to each new customer without the need for the explicit idNumber to be given as input to the constructor.
Note. Assume the classes ShoppingCart and Address are already implemented elsewhere.

Question 3. (20 marks)





According to the above Class diagram, which of the statements below are legal (that is, they do not generate either runtime or compile time errors) and what is the output of each statement, if any.

a) A a = new D();

b) B b = new F();

c) F c = new A();

d) (new E()).c();

e) ((F)(new A())).a();

f) A e = new E();

((E)e).c();

g) D f = new E();

f.a();

a) legal, no output

b) legal, no output

c) illegal

d) legal, output: 5

e) illegal

f) legal, output: 5

g) illegal

Question 4. (20 marks)
Assume Facebook has an internal class called Person that looks like the following class. Each Person is able to keep a list of Person objects, with whom that Person is friends. Every now and then, Facebook makes friendship recommendations based on a simple idea: You could potentially be friends with your friends' friends. In other words, if a and b are friends and also b and c are friends then a and c are also potentially friends based on Facebook's algorithm. Based on this, we can say a and c are potential friends with a separation degree of 1. Now, suppose Facebook has hired you to implement the following two functions given below.

Sample separation degrees:

Separation degree of 1:
a is friends with b
b is friends with c
=> a is potential friends with c

```
class Person {
      private ArrayList<Person> friends = new ArrayList<Person>();
      private String name;

      public Person(String myName){
          this.name = myName;
      }

      public int myCurrentFriendCount(){
          return friends.size();
      }
```

```
public int myPotentialFriendsCountWithOneDegreeOfSeparation()
{
    //you would need to calculate and return the number of
    // potential friends with one degree of separation.
    //In other words, find the total number of friends that the
    // friends of a person have.
    //example: If Alex has two friends Ben and John;
    // and John has two friends; and Ben has five friends;
    // then the number of potential friends with one degree of
    // separation for Alex is 2+5=7.
    //Implement the code here.

    int total = 0;

    for(Person p: friends) {
       total = total + p.myCurrentFriendCount();
    }

    return total;

}
```

```
    public int myPotentialFriendsCountOneDegreeOfSeparationNoDuplicates()
    {
        //you would need to calculate and return the number of
        // potential friends for a Person with one degree of separation
        // with no duplicates.
        //This function is similar to the above with the slight
        // difference that we do not want to count the same friends
        // multiple times.
        //example: If Alex has two friends: Ben and John;
        // and John has two friends; and Ben has five friends;
        // and also John and Ben share one friend,
        // then the number of potential unique friends with one degree
        // separation for Alex is 2+5-1=6.
       //In order to find duplicates, assume that the name instance
       // variable is unique for each Person object.
       //Hint: you may use an ArrayList to keep track of all the unique
       //potential friends of a Person.
        //Implement the code here.

        ArrayList<Person> a = new ArrayList<Person>();

        for(Person p: friends) {
          for( Person p1 : p.friends ) {
            if( a.contains(p1) == false ) {
              a.add(p1);
            }
          }
        }

        return a.size();









    }

     // other methods are omitted to save space
}
```

Hint: The `ArrayList class` has the following method that you can use:
`public boolean contains(Object elem)` - Returns true if this list contains the specified element.

Question 5. (15 marks)
How would you fill in the blank spaces if you knew that the output of the main method was the following?

2
1

Please note that each blank space may include one or more words in it.

```
public class Main {

	public static void main(String[] args){
		D a = new A();
		a.method1();

		E b = new B();
		b.method2();
	}

}
```

```
public ____class_____ F {


	public void method1() {
		System.out.println("2");

	}

}
```

```
public ___abstract class____ E {


	public void method1() {
		System.out.println("4");
	}

	public abstract void method2();

}
```

```
public interface D {


      public void method1();
      public void method2();

}



public class B extends E {


      public void method2() {
            System.out.println("1");
      }

}



public class A extends F implements D {


      public void method2() {
            System.out.println("3");
      }


}
```

Question 6. (10 marks)
What is the output of the following program:

```
public class Sample {

	private int j=0;

	public static void main(String args[])
	{
		int i =5;
		double p =9.5;

		Sample mySample = new Sample();
		Sample myOtherSample = new Sample();

		p = mySample.f(myOtherSample,i);

		System.out.println(p);
		System.out.println(myOtherSample.j);
		System.out.println(mySample.j);


	}

	public double f(Sample s, int i) {
		s.j=i;
		i=9;
		double p=4.95;
		Sample s2 = new Sample();
		s2.j=(int)p;
		s=s2;

		System.out.println(s.j);


		return p;
	}

}
```

Answer:

4
4.95
5
0

Question 7. (15 marks)
Consider the following class A:

```
public class A {

    public int check(String s) {

        int total = 0;

        if (s == null ) {
            throw new IllegalArgumentException();
        }

        return s.length();
    }

}
```

Write the code for the junit test method testCheck()that passes a string “Hello” to the check method and verifies that it works. Write a JUnit test method testInvalidCheck()that tests the check method by providing it with invalid arguments.
Assume that the testCheck() and testInvalidCheck() methods are in class ATest.

```
public class ATest {

    public void testCheck() {
        // Write the code here

          A a = new A();
          int actual = a.check("Hello");
          assertEquals(5, actual);

    }
```

```
    public void testInvalidCheck() {
        // Write the code here

        boolean gotEx = false;
        A a = new A();
        try {
            a.check(null);
        }
        catch(IllegalArgumentException e) {
            gotEx = true;
        }
        assertTrue(gotEx);

    }
}
```

Question 8. (15 marks)
Consider the following classes:

```
public class P {
  public void method2() {
    System.out.println("P 2");
  }
}

public class B extends P {
  public void method2() {
    System.out.println("B 2");
  }

  public void method3() {
    method2();
    System.out.println("B 3");
  }
}

public class C extends B {
  public void method1() {
    System.out.println("C 1");
  }

  public void method2() {
    System.out.println("C 2");
  }
}


public class J extends B {
  public void method1() {
    System.out.println("J 1");
  }

  public void method2() {
    System.out.println("J 2");
  }
}
```

The following variables are defined:

```
B var1 = new B();
P var2 = new J();
B var3 = new C();
B var4 = new J();
Object var5 = new B();
P var6 = new P();
```

In the table below, indicate in the right-hand column the output produced by the statement in the left-hand column. If the statement produces more than one line of output, indicate the line breaks with slashes as in "a / b / c" to indicate three lines of output with "a" followed by "b" followed by "c". If the statement causes an error (compile-time or runtime), fill in the right-hand column with the phrase "error" to indicate this.

Statement	**Output**
`var1.method2();`	B2
`var2.method2();`	J2
`var3.method2();`	C2
`var4.method2();`	J2
`var5.method2();`	error
`var6.method2();`	P2
`var1.method3();`	B2 / B3
`var2.method3();`	error
`var3.method3();`	C2 / B3
`var4.method3();`	J2 / B3

Question 9. (20 marks)
A binary tree is a tree in which each node has at most two child nodes, usually distinguished as "left" and "right". Figure 1 shows a sample binary tree. In this figure,

Node 1 has two children: left child is Node 2 and right child is Node 3.
Node 2 has two children: left child is Node 4 and right child is Node 5.
Node 3 has two children: left child is Node 6 and right child is Node 7.
Nodes 4, 5, 6 and 7 have no children.

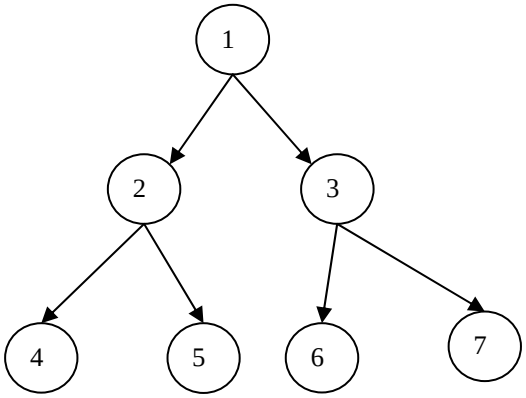


Figure 1.

The main method of class Node first creates the tree (as shown in Figure 1) and then calls the DFS method passing Node 1 as an argument.

```
public class Node {


    private int id;
    private Node left;
    private Node right;



    //Constructor
    public Node(int id, Node left, Node right) {
        this.id = id;
        this.left = left;
        this.right = right;
    }
```

```
    public static void DFS(Node n) {

        System.out.println("Node " + n.id);

        if(n.left != null) {
            DFS(n.left);
        }

        if(n.right != null) {
          DFS(n.right);
        }
    }

    public static void main(String[] args) {

        Node n4 = new Node(4, null, null);
        Node n5 = new Node(5, null, null);
        Node n6 = new Node(6, null, null);
        Node n7 = new Node(7, null, null);
        Node n2 = new Node(2, n4, n5);
        Node n3 = new Node(3, n6, n7);
        Node n1 = new Node(1, n2, n3);

        DFS(n1);
    }

}
```

What is the output when the main method of class Node is executed?

Answer:

Node 1
Node 2
Node 4
Node 5
Node 3
Node 6
Node 7

Question 10. (10 marks)
Consider the following Java code. The classes `E` and `E1` are not checked exceptions. Note that `g()` does not generate any output.

```
1     public class Main {
2       public static void main(String[] args) {
3
4         try {
5           int i = 2;
6           if(________) throw new E();
7           g();
8           int j = 0;
9           i = i / j;
10        }
11        catch(E e) {
12          System.out.println("catch of E");
13        }
14        catch(E1 a) {
15          System.out.println("catch of E1");
16        }
17        catch(ArithmeticException a) {
18          System.out.println("catch of ArithmeticException");
19        }
20        finally {
21          System.out.println("inside finally");
22        }
23
24      }
25
26    }
```

Indicate the output of this code for the following cases:

(a) In Line 6, the Boolean expression is `if (i < 3)` and `g()` does not throw any exception.
Answer:

catch of E
inside finally

(b) In Line 6, the Boolean expression is `if (i == 3)` and `g()` does not throw any exception.
Answer:

catch of ArithmeticException
inside finally

(c) In Line 6, the Boolean expression is `if (i == 3)` and `g()` throws exception `E1`.
Answer:

catch of E1
inside finally