CS 180 Fall 2008 Exam II

There are 20 multiple choice questions. Each one is worth 2 points. There are 3 programming questions worth a total of 60 points.

Answer the multiple choice questions on the bubble sheet given and the programming questions on the exam booklet.

Fill in the Instructor, Course, Signature, Test, and Date blanks. For "Instructor" put your **RECITATION INSTRUCTOR'S LAST NAME GIVEN BELOW**. For "Course" put CS 180. For "Test" put Exam 2.

Fill in the bubbles that correspond to your name, section and Student ID in the bubble sheet. For your section number, use the **SECTION NUMBER** for your lab and project turn in. Consult the following list:

```
0101 THU 07:30 LWSN 1106 Salman Pervez
0201 FRI 07:30 LWSN B134 Cheng Wang
0301 FRI 03:30 LWSN B134 Salman Pervez
0401 FRI 04:30 HAAS G066 Srinivas Pasupuleti
0501 FRI 09:30 LWSN B134 Ashish Gandhe
0601 FRI 02:30 LWSN 1106 Ashish Gandhe
```

For your student ID, use the 10 digit ID number on your student ID card. DO NOT USE YOUR SOCIAL SECURITY NUMBER!

Exams without names will be graded as zero. Only the answers on the bubble sheet will be counted. The questions will be discarded.

Recitation Start Time _____

Recitation TA's Name _____

Student Last Name _____

Student First Name _____

Part I. Multiple Choice Questions (2 points each):

1. Use the code segment below for the first three questions:

```
public class Exam
{
   private int x;

   public Exam() {x = 0;}
   public Exam(int x) {setX(x);}
   public Exam(Exam e) {x = e.getX();}
   public int getX() {return x + 10;}
   public void setX(int x) {this.x = x;}
   public void modifyX(Exam e) {this.x = e.getX() + 10;}

   public static void main(String[] args) {
      Exam e1 = new Exam();
      Exam e2 = new Exam(e1);
      e2.modifyX(e1);
      System.out.println(e1.getX() + "::" + e2.getX());
   }
}
```

   What is the output when the given program is run?

   (a) `20::10`
   (b) `10::20`
   (c) `10::30`
   (d) `30::10`

2. Refer to the code in question 1, what is the output if `x` is declared as `'static'`?

   (a) No change
   (b) `10::10`
   (c) `40::40`
   (d) `30::30`

3. Refer to the code in question 1, what is the output if the method `modifyX` is declared as `'static'`?

   (a) `10::10`
   (b) There is no output due to compile time error
   (c) `30::30`
   (d) `40::40`

4. Which of the following statements about Java's wrapper classes is FALSE?

   (a) They provide utility methods for their primitive counterparts.
   (b) They facilitate information hiding by adding a layer of indirection.
   (c) They have no default contructor.
   (d) They do not provide modifier methods.

5. Assume `array1` and `array2` are both integer arrays with a maximum length of 100. Which of the following statements is correct?

   (a) the Java statement `array2 = array1;` does not cause compile time error.
   (b) the Java conditional expression `(array1 == array2)` tests if the two arrays contain the same elements.
   (c) the Java statement `int len = array1.length();` will assign 100 to the variable `len`.
   (d) the Java statement `array1[100] = 0;` will assign 0 to the last element of the array.

6. What is the output given the following code segment?

```
int[] b = {1, 2, 3, 4, 5};

for (int j = 0; j < b.length; j += 2)
{
   b[j] = b[j--];
   System.out.print(b[j+1]);
}
System.out.println();
```

   (a) `12345`
   (b) `23451`
   (c) The code causes array out of bound error.
   (d) `135`

7. The following method is intended to convert a `String` object into an array.

```
public char[] toArray(String str) {
  int i;
  char[] temp = new char[str.length()];

  for (i = 0; i < str.length(); i++)
    //to do

  return temp;
}
```

   Which statement should replace the line `"//to do"`?

   (a) `temp = temp + str.charAt(i);`
   (b) `temp.setValue(i, str.charAt(i));`
   (c) `temp = str;`
   (d) `temp[i] = str.charAt(i);`

8. What is a valid data type for the following set of data?

```
{ null, {"1"}, {"1", "2"}, {"1", "2", "6"}, {"1", "2", "6", "24"} }
```

   (a) String[]
   (b) String[][]
   (c) int[][]
   (d) int[]

9. Which of the following is FALSE about the `"throws"` clause?

   (a) If a method throws an exception that is not caught within the method, the method ends immediately after the exception is thrown.
   (b) A throws clause in an overriding method can declare to throw an exception different from that which is thrown by the corresponding method in the super-class.
   (c) A method can declare to throw an exception even if that exception is never explicitly thrown from the method.
   (d) A method can declare to throw multiple types of exceptions.

10. What is the output of the following program?

```
public class Parent
{
   private void f()
   {
      System.out.print("parent f()");
   }
   public static void main(String[] args)
   {
      Parent p = new Derived();
      p.f();
   }
}

class Derived extends Parent
{
   public void f()
   {
      System.out.print("derived f()");
   }
}
```

   (a) `parent f() derived f()`
   (b) `parent f()`
   (c) `derived f()`
   (d) `derived f() parent f()`

11. Consider the following class definitions.

```
public class abstract Building{}
public class Office extends Building{}
public class Home extends Building{}
public class House extends Home{}
public class Apartment extends Home{}
```

An object of type `Apartment` also has which of the following types?

(a) `Building`, `Apartment`, and `Home` only
(b) `Apartment` only
(c) `Building`, `Apartment`, `Home` and `Object`
(d) `Apartment` and `Home` only

12. What is the output of the following program?

```
public class Question12
{
   public void method(Object o)
   {
       System.out.print("Object Verion");
   }

   public void method(String s)
   {
      System.out.print("String Version");
   }

   public static void main(String args[])
   {
       Question12 q = new Question12();
       q.method(null);
   }
}
```

(a) `String Version`
(b) There is no output due to null pointer Exception
(c) There is no output due to compile time error
(d) `Object Version`

13. Given the following class declarations, what is the sequence of the constructor calls when a Sandwich object is created?

```
class Meal
{
   Meal() { System.out.print("Meal()"); }
}

class Bread
{
   Bread() { System.out.print("Bread()"); }
}

class Lunch extends Meal
{
   Lunch() { System.out.print("Lunch()"); }
}

class PortableLunch extends Lunch
{
   PortableLunch() { System.out.print("PortableLunch()");}
}

public class Sandwich extends PortableLunch
{
   private Bread b = new Bread();
   public Sandwich()
   {
      System.out.print("Sandwich()");
   }
   public static void main(String[] args)
   {
      Sandwich s = new Sandwich();
   }
}
```

(a) `Bread() PortableLunch() Lunch() Meal() Sandwich()`
(b) `PortableLunch() Lunch() Meal() Bread() Sandwich()`
(c) `Meal() Lunch() PortableLunch() Bread() Sandwich()`
(d) `Sandwich() Bread() Meal() Lunch() PortableLunch()`

14. Which of the following is FALSE about the finally block?

(a) It is legal to throw an exception from within the finally block.
(b) The finally block can be added before a sequence of catch blocks.
(c) The finally block is optional.
(d) The finally block is executed whether or not an exception is thrown.

15. Given that the following try-catch block compiles without errors,

```
try {
  ....
}
catch(AException e) {  }
catch(BException e) {  }
catch(CException e) {  }
catch(DException e) {  }
```

which of the following possible inheritance hierarchies is legal?

(a)
```
AException extends BException
DException extends CException
```

(b)
```
CException extends AException
BException extends DException
```

(c)
```
AException extends CException
BException extends DException
```

(d)
```
BException extends CException
DException extends AException
```

16. Which of the following Java statements is the correct way to open a binary file called `"x.dat"` for writing?

(a) `FileOutputStream x = new ObjectOutputStream("x.dat");`

(b) `ObjectOutputStream x = new ObjectOutputStream("x.dat");`

(c) `ObjectOutputStream x = new ObjectOutputStream(new FileOutputStream("x.dat"));`

(d) `FileOutputStream x = new FileOutputStream(new ObjectOutputStream("x.dat"));`

17. An object's class definition must include which of the following clauses in order to allow object I/O?

(a) implements Portable

(b) implements Recoverable

(c) implements Serializable

(d) implements Writable

18. Which of the following statements is FALSE?

(a) File I/O is used to store data so that we can use them after the program ends.

(b) Not all methods in all classes will throw an EOFException when they try to read beyond the end of a file.

(c) The File class can be used to handle either a text file or a binary file.

(d) A text file typically needs less space than a binary file to store the same numbers.

19. What is the output of the following program?

```
public class Call
{
   private int var;

   public Call (int var) { this.var = var; }

   public void Exchange (Call arr1, Call arr2)
   {
      Call temp = arr1;
      arr1 = arr2;
      arr2 = temp;
   }

   public static void main (String[] args)
   {
      Call[] arr = new Call[3];

      for (int i = 0; i < arr.length; ++i)
         arr[i] = new Call(i + 1);

      arr[0].Exchange(arr[1], arr[2]);

      for (int i = 1; i < arr.length; i++)
         System.out.print(arr[i].var);
   }
}
```

(a) 32

(b) 64

(c) 23

(d) 46

20. Given the following class definitions:

```
public class Base
{
   public void Print ()
   {
      System.out.print("Red");
   }
}

public class FirstDerived extends Base
{
   public void Print ()
   {
      System.out.print("Blue");
   }
}

public class SecondDerived extends Base
{
   public void Print (String str)
   {
      System.out.print("Green");
   }
}
```

What is the output of the following program?

```
public class Dynamic
{
   public static void main (String[] args)
   {
      Base[] b = new Base[2];

      b[0] = new FirstDerived();
      b[1] = new SecondDerived();

      b[0].Print();
      b[1].Print();
   }
}
```

(a) `RedGreen`
(b) `RedRed`
(c) `BlueRed`
(d) `BlueGreen`

Part II. Programming Questions (60 points total):

1. (20 points) You are given the `VendingMachine` interface defined below:

```
public interface VendingMachine
{
    public void vendItem();          //vend one item from the machine
    public int getItemsRemaining();  //get the number of items remaining
    public int getItemsSold();       //get the number of items sold
    public double getCashReceived(); //get the total Cash Received
    public void loadItems(int n);    //load n items into the machine
}
```

Create a `CokeMachine1900` class to represent an old style Code vending machine. It should implement the VendingMachine interface. It should also have the following member variables:

```
int itemsRemaining;
int itemsSold;
double cashReceived;
double unitPrice;
```

The `CokeMachine1900` class should have a single constructor:

```
CokeMachine1900(int nitem, double price)
```

which initializes the variables `itemsRemaining` and `unitPrice`. The other variables have the default value of 0.

When you are finished with the `CokeMachine1900` class, create another class `CokeMachine2000` to represent a newer generation vending machine. `CokeMachine2000` should extend `CokeMachine1900` with the additional method `changeUnitPrice` which updates the `unitPrice`.

Note: Write your code on the next page. It must be a complete class.

Solution for programming question 1:

```
public class CokeMachine1900 implements VendingMachine {
    protected int itemsRemaining;
    protected int itemsSold;
    protected double CashReceived;
    protected double CokePrice;

    public CokeMachine1900(int n, double d) {
        itemsRemaining = n;
        CokePrice = d;
        itemsSold = 0;
        CashReceived = 0;
    }

    public void vendItem() {
        if(itemsRemaining > 0) {
            itemsRemaining--;
            itemsSold++;
            CashReceived += CokePrice;
        }
        else
            System.out.println("Sorry!! Sold Out");
    }

    public int getItemsRemaining() {
        return itemsRemaining;
    }

    public int getItemsSold() {
        return itemsSold;
    }

    public double getCashReceived() {
        return CashReceived;
    }

    public void loadItems(int n) {
        itemsRemaining += n;
    }
}

public class CokeMachine2000 extends CokeMachine1900 {
   public CokeMachine2000(int n, double d) {
      super(n,d);
   }

   public void ChangePrice (double d) {
      CokePrice = d;
   }
}
```

2. (20 points) You have been given the definition of a `Triangle` class below:

```
class Triangle
{
    private int side1;
    private int side2;
    private int side3;

    public Triangle(int s1, int s2, int s3)
    {
        side1 = s1;
        side2 = s2;
        side3 = s3;
    }

    public int getSide1() {return side1;}
    public int getSide2() {return side2;}
    public int getSide3() {return side3;}
}
```

Write a program, call it: `TriangleTest`. This program should include a `main` method and a static method `computeRadius(Triangle t)`. Your program should prompt a user for the 3 sides of a triangle, pass this triangle to `computeRadius()` to compute the radius of a circle which has the same perimeter as that of the triangle, and print out the result if successful.

Your function `computeRadius(Triangle t)` must check for the following two invalid conditions and throw the appropriate exception.

(a) the side of a triangle must be greater than 0

(b) the sum of the lengths of any two sides of a triangle must be greater than the length of the third side

You must define two user-defined exceptions to handle the above errors. These exceptions must inherit from a more general `IllegalTriangleException` which is given.

To summarize, you must write:

- A complete class with the `computeRadius` method and the `main` method. The `main` method should read the input for a `Triangle` and call `computeRadius`.
- Two subclasses of `InvalidTriangleException` to handle the two invalid conditions.

Below is the class definition for: `IllegalTriangleException`

```
class IllegalTriangleException extends Exception
{
    Triangle triangle;
    IllegalTriangleException() {
        super("Illegal triangle parameters");
    }
    IllegalTriangleException(Triangle triangle) {
        this();
        this.triangle = triangle;
    }
}
```

Note: Write your code on the next page. It must be a complete class.

Solution for programming question 2:

```
import java.util.*;

public class TriangleTest {
    public static double computeRadius(Triangle t) throws InvalidSideException,
                                                    TriangleInequalityException {
        int s1, s2, s3;
        s1 = t.getSide1();
        s2 = t.getSide2();
        s3 = t.getSide3();
        if(s1 <= 0||s2 <= 0||s3 <= 0)
            throw new InvalidSideException(t);
        if(s1+s2 <= s3 || s2+s3 <= s1 || s3+s1 <= s2)
            throw new TriangleInequalityException(t);

        return (s1+s2+s3)/(2*Math.PI);
    }

    public static void main(String args[]) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Please enter the three sides");
        Triangle t = new Triangle(scan.nextInt(), scan.nextInt(), scan.nextInt());
        try {
            double radius = computeRadius(t);
            System.out.println("The radius would be: " + radius);
        }
        catch(InvalidSideException e) {
            e.printStackTrace();
        }
        catch(TriangleInequalityException e) {
            e.printStackTrace();
        }
    }
}

class InvalidSideException extends IllegalTriangleException {
    InvalidSideException(Triangle triangle) {
        super(triangle);
    }
}

class TriangleInequalityException extends IllegalTriangleException {
    TriangleInequalityException(Triangle triangle) {
        super(triangle);
    }
}
```

3. (20 points) Write a class called `Reader` which has the ability to read single characters or words from a text file. When there are no more words or characters to be read, the methods responsible for reading should return the String `"NULL"`. Below is the list of the methods your `Reader` should provide:

```
public Reader(String filename) // initialize a Reader object
public void Open()             // open the file for reading
public void Close()            // close the file, no more tokens will be read
public String ReadChar()       // Read the next character
public String ReadWord()       // Read the next word
```

As an example, if your text file, `in.txt`, contains the following text:

```
in
a dark
time
```

And you are running the following program:

```
public static void main(String[] args) {
   Reader reader = new Reader("in.txt");

   reader.Open();

   System.out.println(reader.ReadWord());

   System.out.println(reader.ReadChar());
   System.out.println(reader.ReadChar());
   System.out.println(reader.ReadChar());

   System.out.println(reader.ReadWord());
   System.out.println(reader.ReadWord());

   System.out.println(reader.ReadWord());
   System.out.println(reader.ReadChar());

   reader.close();
}
```

The output should be:

```
in
a
d
a
rk
time
NULL
NULL
```

Recall that `Scanner.next()`, by default, reads and returns the next token in a file separated by whitespace. You may also find the following classes and methods helpful:

```
Scanner: boolean hasNext(), String next(), close()
String:  String valueOf(char), String substring(int), char charAt(int)
```

Note: Write your code on the next page. It must be a complete class.

Solution for programming question 3:

```
import java.util.Scanner;
import java.io.File;
import java.io.FileNotFoundException;

public class Reader
{
    String filename;
    Scanner inStream;
    String nextStr;

    public Reader(String filename) {
        this.filename = filename;
        nextStr = "";
    }

    public void Open() {
        try {
            inStream = new Scanner(new File(filename));
        }
        catch (FileNotFoundException e) {
            System.out.println("File not found");
            System.exit(0);
        }
    }

    private void ReadNext() {
        if(inStream.hasNext()) {
            nextStr = inStream.next();
        }
    }

    public String ReadChar() {
        if(nextStr == "")
            ReadNext();

        if(nextStr == "")
            return "NULL";

        String s = String.valueOf(nextStr.charAt(0));
        if(nextStr.length() == 1)
            nextStr = "";
        else
            nextStr = nextStr.substring(1);

        return s;
    }

    public String ReadWord() {
        if(nextStr == "")
            ReadNext();

        if(nextStr == "")
```

```
            return "NULL";

        String s = nextStr;
        nextStr = "";
        return s;
    }

    public void close() {
        inStream.close();
    }
}
```