Name: _____

Department of Mathematics and Computer Science
CS 1711 MIDTERM TEST
2003 October 24

Read these instructions and **all** of the questions **first**. Do **all** of the questions. **Print** your name at the top of each page **now**. The test is out of 50. Marks are in brackets by the question numbers.

1.[11] a) When might you choose to make a method `private`?

b) What is meant by the the statement that RAM is volatile?

c) What is stored in the *program counter* in the CPU?

d) What three program structures are sufficient to express any algorithm?

e) After the following code executes what is the value of `y`?

```
int x = 27, y;
y = x /( x % 5);
```

2.[4] You have been asked to design the Class `Room`. Required information about a room is the following: its colour, the number of chairs in it, whether the lights are on, and the room's length, width and height dimensions. Write suitable field declarations for the class (and note that it is required that the dimensions be stored in an array of three elements).

3.[4] The following method contains syntax (compile-time) errors. Identify two of them and for each one specify a change that will allow compilation.

```java
public int Mess(int[] age){
  double newAge = 0;
  boolean oldAge;
  for (int i == 0; i < newAge; i++){
    oldAge = age - i;
  }
  return oldAge;
}
```

a) Error:

Fix:

b) Error:

Fix:

4.[10] This question makes use of the `Student` class that is defined on the last page of this test. For each of the following code fragments, write down the output that is produced.

a)
```
Student s1;
s1 = new Student("Billy Bob", -5);
System.out.println("The student's age is:  " + s1.getAge());
```

b)
```
Student s1, s2;
s1 = new Student("Billy Joe", 22);
s2 = s1;
s2.setAge(33);
System.out.println("The student's age is:  " + s1.getAge());
```

c)
```
Student s1, s2;
s1 = new Student("Billy Rae", 44);
System.out.println("The student's age is:  " + s2.getAge());
```

5.[8] Consider a method that has the following header:

```
public int countPositive(int[] arr)
```

This method returns the number of *positive* entries in the array `arr` that is passed as a parameter. Write the `countPositive` method. Use a `for` loop to process the array elements.

6.[7] Consider the following `Point` class, which holds information about a point in the (x, y) plane.

```
public class Point
{
   private double xCoord, yCoord;

   public Point (double inX, double inY)
   {
      xCoord = inX;
      yCoord = inY;
   } // end constructor Point(double,double)

} // end class Point
```

Now suppose you are writing a `Pentagon` class that will contain information about a regular pentagon in the plane (*regular* means that all sides and angles are the same). This class has two fields: a `Point` reference called `centre` that will be used to hold the coordinates of the centre of the pentagon, and a `double` value called `edge` that will hold the length of the edge of the pentagon.

Write a constructor for the `Pentagon` class that takes three `double` parameters — the first is the x-coordinate of the centre, the second is the y-coordinate of the centre, and the third is the length of the edge of the pentagon. Use these values to intialize correctly the fields of the `Pentagon` class. If the edge length passed to the constructor is not positive, set the length of the pentagon's edge to 1.0.

7.[6] A `Pet` class is to be used to keep information about pets and their health. The fields of the class include:

  `isVaccinated`: whether the pet has a current vaccination

  `isYoungest`: whether this pet is the youngest of its litter

  `numSiblings`: the number of other animals in this pet's litter

a) Write an expression that will have the value `true` for a `Pet` object that is neither the youngest nor has been vaccinated, and `false` otherwise. If it is not possible to write such an expression, briefly explain why.

b) Write an expression that will have the value true for a `Pet` object that has a younger sibling, and `false` otherwise. If it is not possible to write such an expression, briefly explain why.

```java
public class Student
{
   private String name;
   private int age;

   public Student(String inName, int inAge) {
      name = inName;
      if (inAge < 0) {
         age = 0;
      }
      else {
         age = inAge;
      }
   } // end constructor Student(String,int)

   public String getName() {
      return name;
   } // end getName()

   public int getAge() {
      return age;
   } // end getAge()

   public void setName(String newName) {
      name = newName;
   } // end setName(String)

   public void setAge(int newAge) {
      if (newAge < 0) {
         age = 0;
      }
      else {
         age = newAge;
      }
   } // end setAge(int)
} // end class Student
```