Last Name	
First Name	
Student Number	

Instructions:

(a) There are 4 questions on this test . The test is worth 35% of your final mark.
(b) There is a time limit of 50 minutes.
(c) Write your answers on this test in pen only. (you may write on the back if space is needed)
(d) Make sure you put your name and student # on everything
(e) Clearly identify which question you are answering.
(f) Please include comments to explain what you are doing.
(g) There are 8 versions of this test randomly distributed in the class…Do not attempt to copy from another paper.

Question 1a) (5 marks) Rewrite the following do loop into a while loop.

```
int n = 1;
double x = 0;
double s;
do
{
     s = 1.0 / (n *n);
     x = x+s;
     n++;
} while (s>0.01);
```

Question 1b) (5 marks) Rewrite the following for loop into a while loop.

```
for(i=0;i<50;i++);
```

Question 2) (5 marks) Given the following variable definitions,

```
int v1,v2,v3;
```

Ask the user to input three integers (one at a time). Read and store the input in each of these three variables. Your program should then "sort" the variables so that when the program ends the smallest value that was input will be in ***v1***, the middle value input should be in ***v2*** and the largest value will be in ***v3***. Print out ***v1***, ***v2*** and ***v3***. You can make more variables if you need them. You need not write a whole class, just write the code required to do this task.

An example run might look like… (note: this is just an example)

```
Enter an integer.
5
Enter an integer
2
Enter an integer
1
The sorted list is
1 2 5
```

Question 3) Given the following correct Java class,

```
class Digby
{
        private int muscle_count;

        public Digby(int num_muscles){
                muscle_count = num_muscles;
        }

        public Digby(){
                muscle_count = 100;
        }

        public void harvest(int how_many){
                muscle_count = muscle_count-how_many;
                if(muscle_count < 0)muscle_count = 0;
        }

        public int howmany(){
                return muscle_count;
        }

        public static void message(){
                System.out.println("This is a message.");
}}
```

Answer the questions on the following page

Question 3 (continued from previous page)

a) (2 marks) Write a single line of java code that will make a variable called ***janet*** of type ***Digby*** and make a new object that has ***a muscle_count*** of 13.

b) (1 mark) How many class variables are in the class ***Digby***?

c) (2 marks) Write a line of Java code that will print out the current muscle_count of an object of type Digby called ***kermit***.

d) (5 marks) Write another instance method called ***goof*** that takes a single argument and returns the argument multiplied by ***muscle_count***.

Question 4 (10 marks) Implement a class ***Truck*** with the following properties. A truck has a certain fuel efficiency measured in km/liter, and a certain amount of fuel in the gas tank. The efficiency is specified in the constructor, and the initial fuel level is 0. Supply a method ***drive*** that simulates driving the truck for a certain distance, reducing the fuel level in the gas tank. Also supply methods ***getGas***, returning the current fuel level, and ***addGas***, to tank up. The outline of the class is given below -- simply fill in the parts labeled "**todo**" in the answer space after the code below. Note: A truck has a fuel level and an efficiency.

```
public class Truck
{
    //instance variables
    todo_a_2_mark
    /**
          Constructs a truck with a zero fuel and given
          efficiency @param kmPerLiter */

    public Truck(double kmPerLiter)
    {
          fuel = todo_b_1_mark
          efficiency = todo_c_1_mark
    }

    /**
       Drives the truck the given distance
       @param distance */
    public void drive(double distance)
    {
          todo_d_2_marks
    }

    /**
       Puts fuel in the truck.
       @param amount the amount of gas added */
    public void addGas(double amount)
    {
          todo_e_2_marks
    }

    /**
       Gets the current fuel level of the truck.
       @return the current fuel level*/
    public double getGas()
    {
          todo_f_2_marks
    }
}
```

answer space below

Your code for **todo_a_2_mark** goes in this box

Your code for **fuel = todo_b_1_mark** goes in this box

Your code for **efficiency = todo_c_1_mark** goes in this box

Your code for **todo_d_2_marks** goes in this box

Your code for **todo_e_2_mark** goes in this box

Your code for **todo_f_2_mark** goes in this box