**GENE 121** **Assignment 7** **Winter 2013**

---

This assignment will be **completed in pairs**.

At the start of the lab session you should sit at your assigned box of Lego parts corresponding to your team number. All parts should be returned at the end of the lab session.

---

In this assignment you will be reviewing:

- Decisions
- Loops
- Arrays
- Encoders, buttons, timers, and colour sensor

And using:

- Functions

Don't forget to include a signed header for your assignment, and to make sure that both students in the group have an electronic copy of the code when it is complete.

**Question 1 (RobotC)**

a) Write a function that waits for a button to be pressed and released, and then returns the number of the button that was pushed.

b) Write a function that drives the robot in a square going either clockwise, or counter clockwise. The function must receive a boolean variable indicating direction of travel and any other parameters you may find useful.

c) Write a main that calls the function from (a) and does one of the following things:
   - If the left button is pressed, the robot should drive the square counter-clockwise (by calling the function from (b))
   - If the right button is pressed, the robot should drive the square clockwise (by calling the function from (b))
   - If the orange button is pressed, the program should quit

   At the end of the program, you should display the day and your group number.

   Demonstrate your program to a TA and then print and submit the code.

**Question 2 (RobotC)**

In this question you are going to experimentally determine the number of motor counts that the motors rotate when the entire robot turns 1 degree. You will do this by placing an obstacle at the starting point that the robot can "see" using the ultrasonic sensor. This obstacle will indicate that one full revolution of the robot has taken place.

- Write a function that spins the robot in place (i.e. one motor forwards and one motor backwards) until it sees the obstacle, and then returns the encoder count of the motor.
- Write a main program that will,
  - call the above function once to set the starting position; (Recall that the robot's field of view is $\approx 30°$ so the starting position is not when the robot is directly facing the object.)
  - call the above function 4 times, and display the encoder count for each revolution on a separate line of the display
  - average the results, and print to the display the average number of encoder counts of motor rotation per degree of robot rotation
  - include your day and group number on the display.

Demonstrate your program to a TA. Print your code, and **copy the screen output** onto your printout. Submit the code and output. **Also, state where and how this number would be useful.**

**Question 3 (Dev-C++)**

A landscaping company has a crew of co-op students who do both paving and fencing. The company charges $18.50 for paving each square metre, or any part of a square metre, (i.e. 2.1 becomes 3 sq. m) and $26.00 per metre, or any part of a metre, for fencing. All fences also have a $120.00 gate. Added to each bill is a $35.00 administration fee.

The company keeps a list of its top 10 customers, whose names are stored in "**`prefer.dat`**". These customers are given a 10% discount on the total before taxes.

The sales representative for the company has created a file called "**`jobs.dat`**" that contains the jobs completed by the work crew for the month of July. Each line of the file contains the following information:

- Job number (integer)
- Customer name
- Paving Job? (Boolean indicating whether paving has been done)
- Fencing Job? (Boolean indicating whether fencing has been done)
- Shape (string indicating the shape of the area paved/fenced)
- Dimensions (varying number of distances depending on the shape)

The shape names are all lower case (as seen in the table below):

| Actual Name | Shape | Dimensions |
|---|---|---|
| `tri` | Triangle | 3 sides |
| `rect` | Rectangle | Length and width |
| `quad` | Quadrilateral | Side1, side2, diagonal, side3, side4 |
| `sect` | Sector | Radius, angle in degrees |

Typical lines of the file "**`jobs.dat`**" are:

```
7100  Rennick    1     1     tri   3     4     5
7101  Lowe       1     0     rect  3     7
7102  Yiu        1     0     sect  3     30
```

Write a program to do the calculations and produce a printout similar to the following:

```
Job #   P Area   P Cost    F Length   F Cost    Tax    Prefer    Total Cost
  ..      ..       ..         ..        ..       ..      ..          ..
```

<u>You may assume a tax rate of 13%.</u>

You must write the following functions:

1. A function which returns the area of a triangle. **<u>This function must also be used for finding the area of a quadrilateral</u>**. Use the following function prototype:
   - `double areaTri (double, double, double);`
2. A function which receives the length and width of a rectangle, and calculates and returns the area. Use the following function prototype:
   - `double areaRec(double L, double W);`
3. A function which receives the length and width of a rectangle, and calculates and returns the perimeter. Use the following function prototype:
   - `double perimRec(double L, double W);`
4. Similar perimeter and area functions for the other shapes (quad and sect)
5. Well-named functions(s) to calculate and print costs. For ease of debugging, print a column indicating whether the customer is preferred or not.

Submit your program and output for the files given on UW-Learn.