MIDTERM EXAM

EMPLID

CSCI 135

NAME: FIRST LAST

1. (12%) Suppose your program has the following declarations to represent information about a student:

```
string major ; // possibly empty
float gpa ;
bool female ;  // true if female , false if male
```

Write C++ logical conditions corresponding to each of the following sets. Your answers should be as compact
as possible and cover all cases.

(a) Female computer science majors with GPAs between 3.5 and 3.9.

```
female && major == "computer science" && gpa > 3.5 && gpa < 3.9
```

(b) Male students, whose major starts with the letter 'e' (economics, english, etc), and whose GPA is 2.0 or lower.

```
!female && major[0] == 'e' && gpa < 2.0
```

(c) All students, whose major ends in the letter 's' (mathematics, physics, etc), and whos GPA is a perfect 4.0.

```
major[major.length() - 1] == 's' && gpa == 4.0
```

2. (10%) Write a C++ function that calculates: $\sqrt{\frac{137(x-y)}{z^{n-1}}}$

```
#include <cmath>
double foo(double x, double y, double z, double n)
{
  return sqrt(137 * (x - y) / pow(z, n - 1));
}
```

3. (18%) Consider the following program fragment:

```
int enigma (int a, int & b);

int main() {
    int x = 0; // "SPECIAL LINE"
    cout << x++;
    cout << ++x << endl;
    for (int k = 1; k < 3; k++)
        cout << enigma(k , x);
    return 0 ;
}

int enigma(int a, int & b) {
    static int c = 0;
    c = a++;
    b += 2;
    return c * b;
}
```

(!!! this is a dashed box

(a) What does the program output?

0 2
4 12

(b) Circle all actual arguments in the program.

(c) Underline all formal parameters in the program.

(d) Draw a dashed box around all prototypes in the program.

(e) Draw a solid box around the scope of the variable declared on SPECIAL LINE?

(f) What is the value of variable `c` at the end of program execution - just before the `main()` function returns?

2

4. (15%) Write a function: `void average_word_lenght(string & sentence, float & result)` that calculates the average length of all words in the string `sentence`.

```
#include <string>
#include <iostream>
using namespace std;

void average_word_lenght(string & sentence, float & result) {
    float sum = 0;
    int count = 1;
    for (int i = 0; i < sentence.length(); i++)
    {
        if (sentence[i] == ' ') {
            count++;
        }
        else {
            sum++;
        }
    }
    result = sum / count;
}

int main()
{
    string sentence = "";
    float result;
    average_word_lenght(sentence, result);
    cout << result << endl;
    return 0;
}
```

5. (15%) Write a program that asks user for a positive integer side length. If they enter an illegal value, they must be prompted to enter a good one until they do. It then displays, using asterisks, a filled diamond of the given side length. For example, if the side length is 4, the program should display:

```
   *
  ***
 *****
*******
 *****
  ***
   *
```

```
int main()
{
   cout << "Enter the length of the diamond side: ";
   int side;
   cin >> side;
   // Calculate the max width of the diamond
   int max_width = (side * 2) - 1;
   // Calculate half of that to place spaces
   int half_width = (max_width) / 2;
   int dots = 1;
   // Print top half of diamond
   for (int i = 0; i < side; i++)
   {
      // Print spaces to line up triangle
      for (int j = 0; j < half_width; j++)
      {
         cout << " ";
      }
      // Print dots
      for (int j = 0; j < dots; j++)
      {
         cout <<"*";
      }
      cout << endl;
      dots = dots + 2;
      half_width--;
   }
   // Reset variables for bottom half
   dots = max_width - 2;
   half_width = 1;
   // Print bottom half of diamond
   for (int i = 0; i < side; i++)
   {
      // Print spaces to line up triangle
      for (int j = 0; j < half_width; j++)
      {
         cout << " ";
      }
      // Print dots
      for (int j = 0; j < dots; j++)
      {
         cout <<"*";
      }
      cout << endl;
      dots = dots - 2;
      half_width++;
   }
   return 0;
}
```