AM 3611: C++ for Scientific Computing

Assignment 1: Intro, Flow Control

Due: 17 September 2018

Yizhou Tang

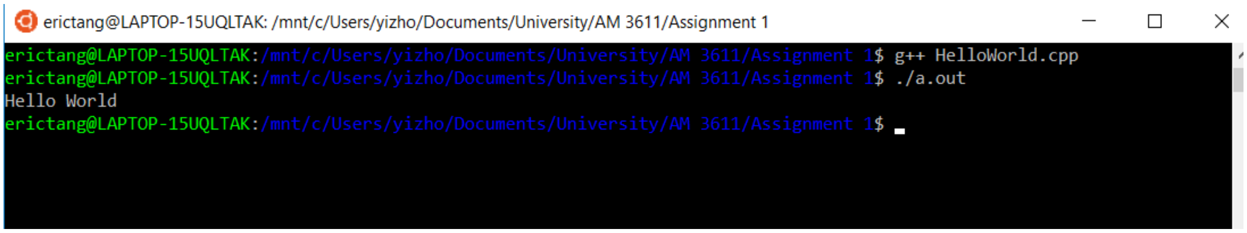
250888541

Question 1.1:

i) Code for 1.1:

```
#include <iostream>

int main(int argc, char* argv[])

// Question 1.1 from the text.

// This program prints out "Hello World" to the screen

{

        std::cout << "Hello World\n";

        return 0;

}
```

ii) Screenshot of the program:



erictang@LAPTOP-15UQLTAK: /mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1

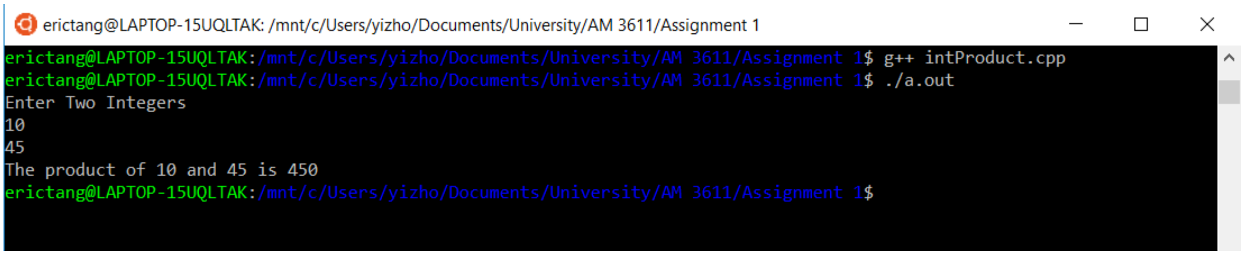
```
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ g++ HelloWorld.cpp
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ ./a.out
Hello World
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ _
```

Question 1.2:

i) Code for 1.2:

```
#include <iostream>

int main(int argc, char* argv[])

// Question 1.2 from the text

// This program asks the user for two integer inputs
// then it writes the product of these integers to the screen

{

	int num1, num2, result;

	std::cout << "Enter Two Integers" << std::endl;

	std::cin >> num1;

	std::cin >> num2;

	result = num1 * num2;

	std::cout << "The product of " << num1 << " and " << num2 <<" is " << result <<"\n";

	return result;

}
```

ii) Screenshot of the program:



erictang@LAPTOP-15UQLTAK: /mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1

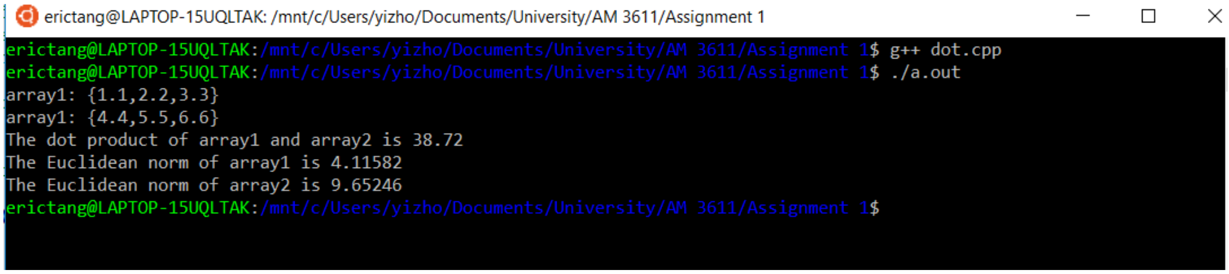
```
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ g++ intProduct.cpp
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ ./a.out
Enter Two Integers
10
45
The product of 10 and 45 is 450
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$
```

Question 1.3:

i) Code for 1.3:

```
#include <cmath>

#include <iostream>

int main (int argc, char* argv[])

//Question 1.3 from the text

//This program declares two vectors as arrays of double precision floating point numbers

// of length 3 and assigns values to each of the entries.

// It calculates the dot product of these vectors and prints it to screen.

// Lastly, it also prints the Euclidean norm of both vectors to screen.

{

	//Declare arrays of double precision floating point numbers of length 3
	double array1[3] = {1.1,2.2,3.3};

	double array2[3] = {4.4,5.5,6.6};

	// Compute the dot product of the two arrays

	double dotProduct;

	dotProduct = (array1[0] * array2[0]) + (array1[1] * array2[1]) + (array1[2] * array2[2]);


	// Print out the results

	std::cout<< "array1: {" << array1[0] << "," << array1[1] << "," << array1[2] << "}" << std::endl;

	std::cout<< "array1: {" << array2[0] << "," << array2[1] << "," << array2[2] << "}" << std::endl;

	std::cout << "The dot product of array1 and array2 is " << dotProduct << std::endl;


	// Euclidean norm of both vectors

	double norm1;

	double norm2;

	norm1 = sqrt(pow(array1[0],2) + pow(array1[1],2) + pow(array1[2],2));

	norm2 = sqrt(pow(array2[0],2) + pow(array2[1],2) + pow(array2[2],2));

	std::cout<<"The Euclidean norm of array1 is " << norm1 << std::endl;

	std::cout<<"The Euclidean norm of array2 is " << norm2 << std::endl;

	return 0;

}
```

ii) Screenshot of the program:



erictang@LAPTOP-15UQLTAK: /mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1

```
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ g++ dot.cpp
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ ./a.out
array1: {1.1,2.2,3.3}
array1: {4.4,5.5,6.6}
The dot product of array1 and array2 is 38.72
The Euclidean norm of array1 is 4.11582
The Euclidean norm of array2 is 9.65246
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$
```

Question 1.4:

i) Written description for 1.4:

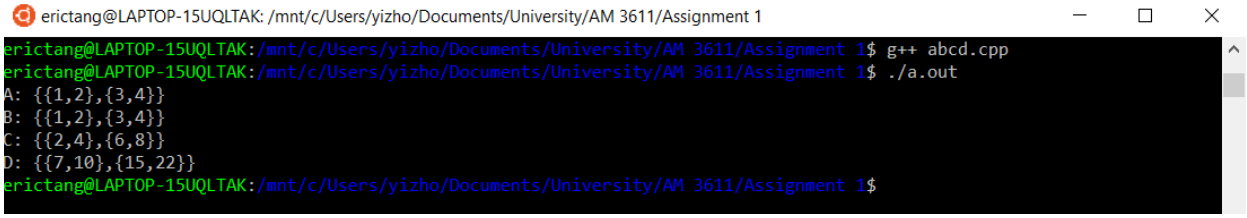
The code for this question declares four 2v2 matrices of double precision floating point numbers: A, B, C, and D. The values for A are assigned as {{1, 2}, {3, 4}}, the values for B are assigned as {{1, 2}, {3, 4}} as well. After declaring the variables, the code computes the values for C and D. C = A + B, it is computed by following the matrix addition rules. D = AB, it is computed by following the matrix multiplication rules.

ii) Code for 1.4:

```
#include <cmath>
#include <iostream>
int main (int argc, char* argv[])
// Question 1.4 from the text
// This program declares four 2x2 matrices of double precision floating point numbers:
// A,B,C,D, and assigns values to the entries of A and B. It then calculates C and D.
// C = A + B
// D = AB
// Lastly it prints the entries of these matrices to screen

{
        //Declare arrays of double precision floating point numbers of length 3
        double A[2][2] = {{1.0,2.0},{ 3.0,4.0}};
        double B[2][2] = {{1.0,2.0},{ 3.0,4.0}};
        double C[2][2];
        double D[2][2];

        //Compute C
        C[0][0] = A[0][0] + B[0][0];
        C[0][1] = A[0][1] + B[0][1];
        C[1][0] = A[1][0] + B[1][0];
        C[1][1] = A[1][1] + B[1][1];

        //Compute D
```

```
    D[0][0] = (A[0][0] * B[0][0]) + (A[0][1] * B[1][0]);
    D[0][1] = (A[0][0] * B[0][1]) + (A[0][1] * B[1][1]);
    D[1][0] = (A[1][0] * B[0][0]) + (A[1][1] * B[1][0]);
    D[1][1] = (A[1][0] * B[0][1]) + (A[1][1] * B[1][1]);


    // Print out the results
    std::cout<< "A: {{" << A[0][0] << "," << A[0][1] << "},{" << A[1][0] << "," << A[1][1] << "}}" << std::endl;
    std::cout<< "B: {{" << B[0][0] << "," << B[0][1] << "},{" << B[1][0] << "," << B[1][1] << "}}" << std::endl;
    std::cout<< "C: {{" << C[0][0] << "," << C[0][1] << "},{" << C[1][0] << "," << C[1][1] << "}}" << std::endl;
    std::cout<< "D: {{" << D[0][0] << "," << D[0][1] << "},{" << D[1][0] << "," << D[1][1] << "}}" << std::endl;



    return 0;
}
```

iii) Screenshot of the program:



erictang@LAPTOP-15UQLTAK: /mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1

```
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ g++ abcd.cpp
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ ./a.out
A: {{1,2},{3,4}}
B: {{1,2},{3,4}}
C: {{2,4},{6,8}}
D: {{7,10},{15,22}}
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$
```

iv) Comments:

From (iii) we can see that the program gave {{2, 4}, {6, 8}} as the sum of {{1, 2}, {3, 4}} and {{1, 2}, {3, 4}}. The result shows the algorithm is following the correct rules for matrix addition:

A = {{1, 2}, {3, 4}}

B = {{1, 2}, {3, 4}}

C = A + B

C = {{1+1, 2+2}, {3+3, 4+4}}

C = {{2, 4}, {6, 8}}

From (iii) we can also see that the program gave {{7, 10}, {15, 22}} as the product of AB.

The result shows the algorithm is following the correct rules for matrix multiplication:

D = AB

D = {{(1*1+2*3), (1*2+2*4)}, {(3*1+4*3), (3*2+4*4)}}

D = {{7, 10}, {15, 22}}

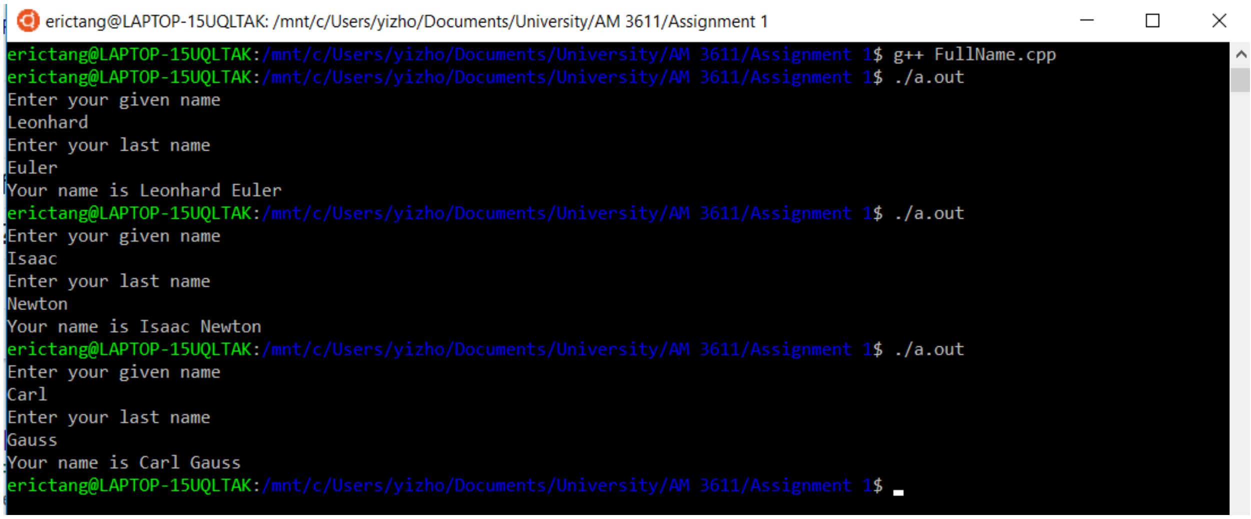
Question 1.5:

i) Code for 1.5:

```
#include <iostream>

#include <string>

int main(int argc, char* argv[])

// Question 1.5 from the text

// This program asks the user for two separate strings (given name and family name)

// then prints the user's full name to the screen

{

        std::string givenName;

        std::string familyName;

        //Asks user for given name

        std::cout << "Enter your given name" << std::endl;

        //Save the input into the varaible "givenName"

        std::cin >> givenName;

        //Ask user for last name

        std::cout << "Enter your last name" << std::endl;

        //Save the input into the variable "lastName"

        std::cin >> familyName;

        //Print out the full name to the screen

        std::cout<< "Your name is " << givenName << " " << familyName << std::endl;

        return 0;
```

}

ii) Screenshot for the program:



erictang@LAPTOP-15UQLTAK: /mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1

```
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ g++ FullName.cpp
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ ./a.out
Enter your given name
Leonhard
Enter your last name
Euler
Your name is Leonhard Euler
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ ./a.out
Enter your given name
Isaac
Enter your last name
Newton
Your name is Isaac Newton
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ ./a.out
Enter your given name
Carl
Enter your last name
Gauss
Your name is Carl Gauss
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$
```

Question 1.6:

i) Written description for 1.6:

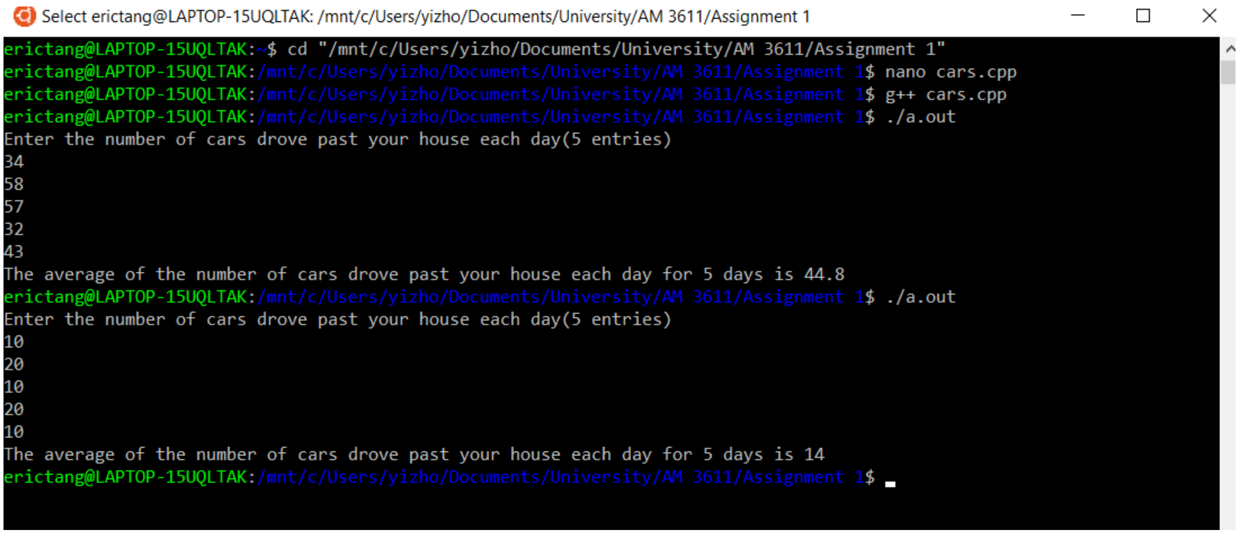
The code for this question invites the user to enter 5 different integers and returns the average of the inputs. Before the average calculation, the sum of the 5 inputs are converted to double precision floating point numbers to avoid integer division.

ii) Code for 1.6:

```
#include <iostream>

int main(int argc, char* argv[])

//This program records 5 inputs(number of cars drove past the user's house)from the user into an integer array

// then calculates and prints out the average of these 5 integers

{

        //Declare an empty array for user inputs

        int carNums[5];

        int sum;

        double result;

        //Ask the user for inputs and record them into the array

        std::cout << "Enter the number of cars drove past your house each day(5 entries)" << std::endl;
```

```
            std::cin >> carNums[0];
            std::cin >> carNums[1];
            std::cin >> carNums[2];
            std::cin >> carNums[3];
            std::cin >> carNums[4];
            //Calculate the sum of all 5 numbers
            sum = carNums[0] + carNums[1] + carNums[2] + carNums[3] + carNums[4];
            //Divide the sum by 5 to get the average
            //Use C++ static_cast to convert "sum" to a double
            result = static_cast<double> (sum) / 5;
            //Print out the final statement
            std::cout << "The average of the number of cars drove past your house each day for 5 days is "
<< result << std::endl;
}
```

iii) Screenshot of the program:


Select erictang@LAPTOP-15UQLTAK: /mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1

```
erictang@LAPTOP-15UQLTAK:~$ cd "/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1"
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ nano cars.cpp
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ g++ cars.cpp
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ ./a.out
Enter the number of cars drove past your house each day(5 entries)
34
58
57
32
43
The average of the number of cars drove past your house each day for 5 days is 44.8
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ ./a.out
Enter the number of cars drove past your house each day(5 entries)
10
20
10
20
10
The average of the number of cars drove past your house each day for 5 days is 14
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$
```

iv) Comments

From (iii) we can see that the program has worked as expected. Using the sample data 34, 58, 57, 32, 43, it successfully produced the correct answer as suggested by the textbook. Another sample set was used to check the program: 10, 20, 10, 20, 10. The program successfully computed the correct answer as well, since (10+20+10+20+10)/5 = 14.

Question 1.7:

i) Code for 1.7:

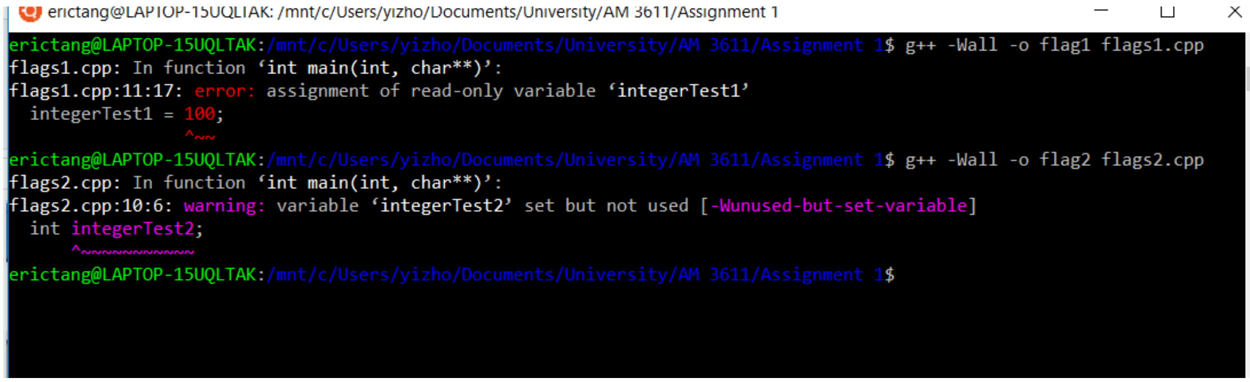
1. Attempt to change a constant integer variable's value：

```
#include <iostream>
int main(int argc, char* argv[])
//Question 1.7 from the text
//This program is meant to be used to
//investigate the use of the compiler error warning flags
// 1. Attempt to change a constant integer variable's value
{
        //Declare the constant variable
        const int integerTest1 = 10;
        //Attempt to change integerTest1's value
        integerTest1 = 100;
}
```

2. Attempt to set an integer's value to 3.2:

```
#include <iostream>
int main(int argc, char* argv[])
//Question 1.7 from the text
//This program is meant to be used to
//investigate the use of the compiler error warning flags
// 2. Attempt to set an integer's value to 3.2

{
        //Declare the integer variable
        int integerTest2;
        //Attempt to set integerTest2 to the value 3.2
        integerTest2 = 3.2;
}
```

ii) Screenshot of the programs:



erictang@LAPTOP-15UQLTAK: /mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1

```
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ g++ -Wall -o flag1 flags1.cpp
flags1.cpp: In function ‘int main(int, char**)’:
flags1.cpp:11:17: error: assignment of read-only variable ‘integerTest1’
  integerTest1 = 100;
                 ^~~
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$ g++ -Wall -o flag2 flags2.cpp
flags2.cpp: In function ‘int main(int, char**)’:
flags2.cpp:10:6: warning: variable ‘integerTest2’ set but not used [-Wunused-but-set-variable]
  int integerTest2;
      ^~~~~~~~~~~~
erictang@LAPTOP-15UQLTAK:/mnt/c/Users/yizho/Documents/University/AM 3611/Assignment 1$
```