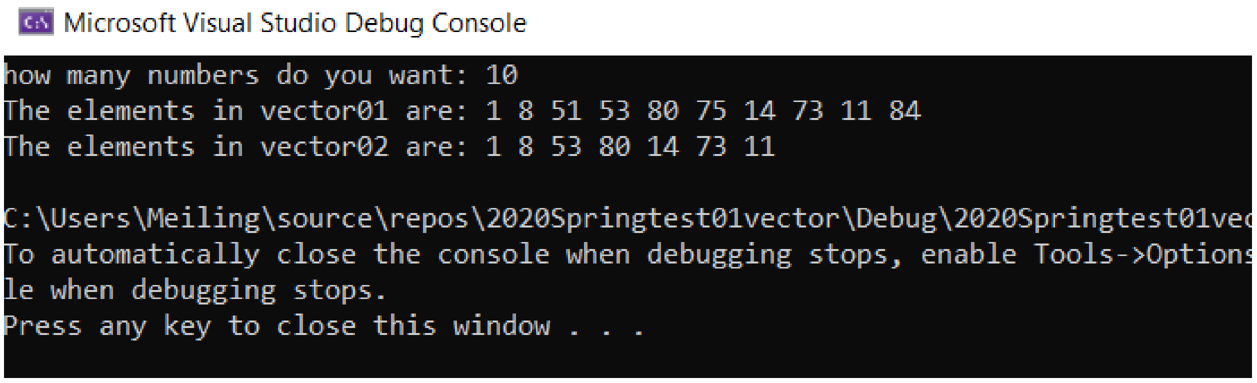
MTRE 2610 Engineering algorithms and visualization

Test 01 – Fall 2020– Meiling Sha

This test is open notes, open book, open computer, and open Internet. However, you may not communicate with another human in any way during the test. Upload a single ZIP file to D2L consisting of a folder named Test01Lastname, itself containing a single folder of the same name, inside which are three folders, one for the Visual Studio project of each problem.

1. Write a program to request the user enter an integer. Then create a vector with that number of elements between 1 and 100 randomly. Then create another vector containing only the elements of the first vector which are **NOT** multiples of 3, and display the elements in both vectors to the screen. For example:



Microsoft Visual Studio Debug Console

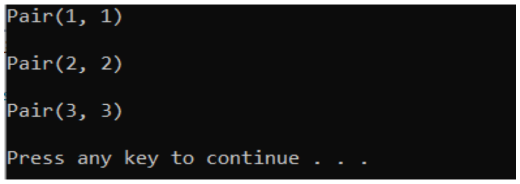
```
how many numbers do you want: 10
The elements in vector01 are: 1 8 51 53 80 75 14 73 11 84
The elements in vector02 are: 1 8 53 80 14 73 11

C:\Users\Meiling\source\repos\2020Springtest01vector\Debug\2020Springtest01vec
To automatically close the console when debugging stops, enable Tools->Options
le when debugging stops.
Press any key to close this window . . .
```

2. Create a class called IntPair that holds two integers. This class should have two member variables to hold the integers. You should also create members functions: two constructors, and one named "set" that will let you assign values to the integers, and one named "get" that will allow you to access values to the integers, and overload operator + as a member function to add two IntPair objects by adding their two data members respectively, then overload the insertion operator << to output the object of IntPair class.

The following main function should execute:

```
int main()
{
        IntPair p1,p3;

        p1.set(1, 1); // set p1 values to (1, 1)
        IntPair p2( 2, 2 }; // initialize p2 values to (2, 2)

        p3=p1+p2;

        Cout<<p1<,endl;
        Cout<<p2<<end;
        Cout<<p3<<endl;

        return 0;
}
```

and produce the output:



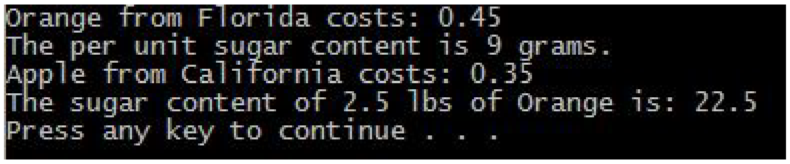
```
Pair(1, 1)

Pair(2, 2)

Pair(3, 3)

Press any key to continue . . .
```

3. Using the provided class Produce(you can find it on D2L) , create a new class that will inherit from Produce. The child class will be called Fruit. This class will have the following data member: sugarContent. The class will also have its own display function to also display the per unit sugarContent while calling the display function from the parent (see the output), and member function getTotalSugars that will return the total sugar content based on the following equation:

sugarContent * pounds

Use the following main function to achieve the desired output:

```
#include <iostream>
#include "Produce.h"

using namespace std;

int main() {
	Fruit orange("Orange", "Florida", 0.45, 9);
	Produce apple("Apple", "California", 0.35);
	double sugars;
      double pounds=2.5;
	orange.display();
	cout << apple << endl;

	sugars = orange.getTotalSugars(pounds);
	cout << "The sugar content of"<<pounds<< "lbs of " << orange.getName() << " is: "
<<sugars << endl;

	system("pause");
	return 0;
}
```



```
Orange from Florida costs: 0.45
The per unit sugar content is 9 grams.
Apple from California costs: 0.35
The sugar content of 2.5 lbs of Orange is: 22.5
Press any key to continue . . .
```