Module Code: COS1512

Assessment: Assignment 4

Student Number: 69234175

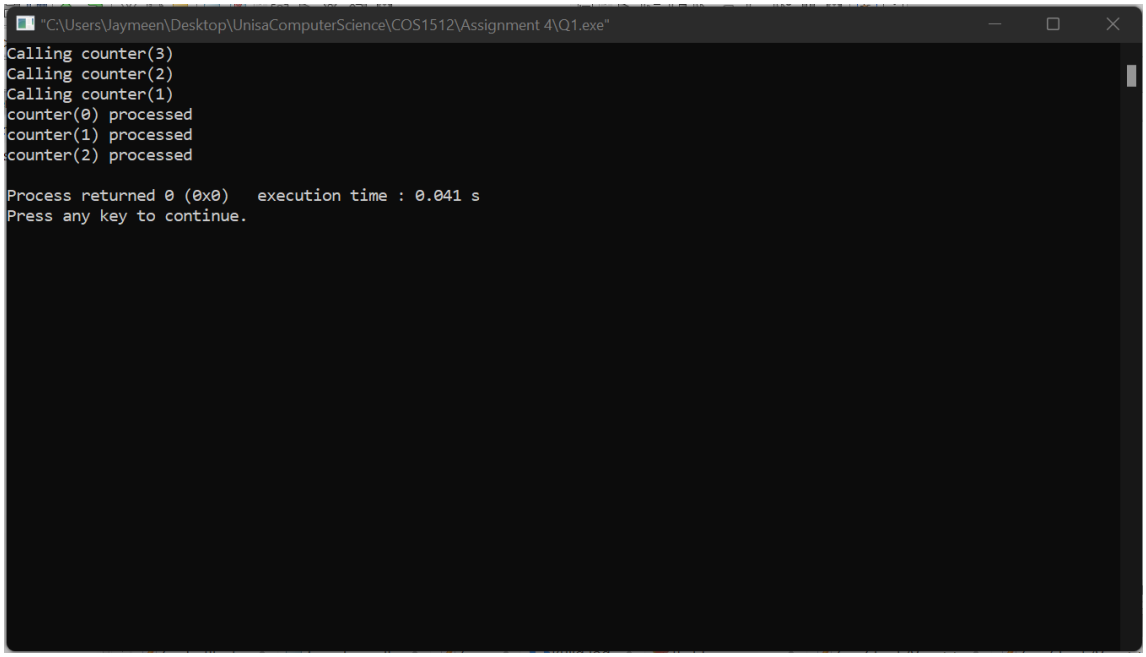
Name: Jaymeen Patel

Unique Number: 170516

Due Date: 19/09/2022

Question 1

a)



b) The purpose of a base case is to end a recursive function and not let it run infinitely.

c) The general case is where the recursive function is called.

d) The base case is, if (counter == 0)

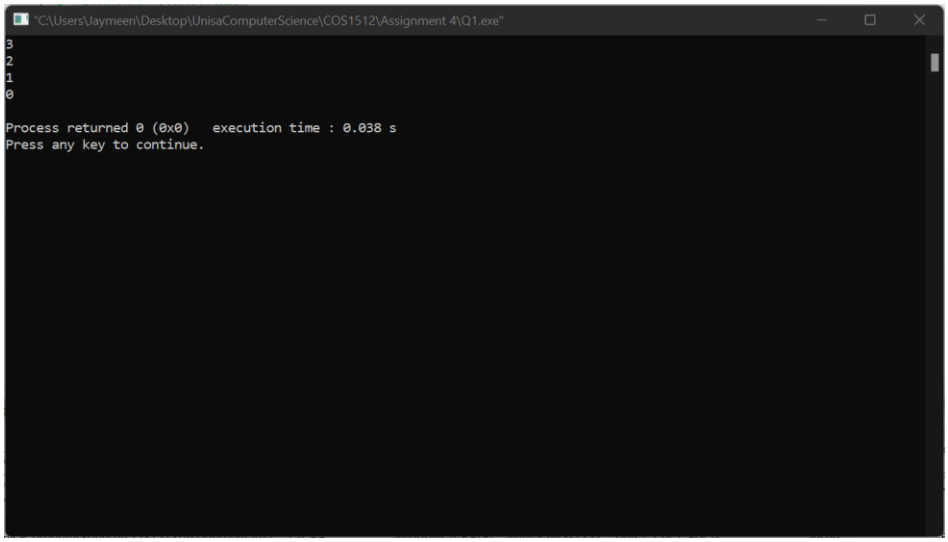
e) The general case is, count(--counter);

f) #include <iostream>

```
using namespace std;
void count(int counter)
{

if (counter == -1)
return;
else
{
cout << counter << endl;
count(--counter);

return;
}
}
int main()
{
int i = 3;
count(i);
return 0;
}
```



Question 2

a) Line 18 is not a valid access because the getter/accessor is protected and not public.

b) Line 32 is a valid statement because the object for function A is declared correctly.

c) Line 42, 42: int a = objA.getX();

d) Class C private variables : none

protected variables : int getx();

int z

public variables : void setx();

int getZ();

void setZ();

e) Class C private variables : none

protected variables : int getx();

int z

void setx();

public variables :

int getZ();

void setZ();

Question 3

```
#include <iostream>
#include<string>


using namespace std;


class Marks
{

public:
 Marks();

 Marks (string name, string number, int asg1, int asg2, int asg3, double test);

 double calcMark() const;

 string getName() const;

 string getNumber() const;

 double getassignmentMarks()const;




private:
 string stdtName;
 string stdNumber;
 int assignments [3];
 double testMark;
```

```
};

Marks::Marks()
{
   stdtName = "Eric";

   stdNumber = "654653176 ";

   assignments[0] = 67;

   assignments[1] = 74;

   assignments[2] = 60;

   testMark = 77;

}


Marks::Marks(string name, string number, int asg1, int asg2, int asg3, double test)

{
   stdtName = name;

   stdNumber = number;

   assignments[0] = asg1;

   assignments[1] = asg2;

   assignments[2] = asg3;
```

```
    testMark = test;
}



double Marks::calcMark() const
{
    return testMark;
}




string Marks::getName() const
{
    return stdtName;
}




string Marks::getNumber() const
{
    return stdNumber;
}



double Marks:: getassignmentMarks() const
{
    return(assignments[0] + assignments[1] + assignments[2]);
}
```

```
////
////
class FinalMark : public Marks
{

 double eMark;

 public:

FinalMark(): Marks{}
{
   eMark = 78;
}

FinalMark(string name, string number, int asg1, int asg2,
        int asg3, double test, double exMark) : Marks{name, number, asg1, asg2, asg3, test}
{
   eMark = exMark;
}



int calcMark() const
{
   double asgMarks, test, fmark;

   test =  Marks::calcMark();

   asgMarks = (Marks::getassignmentMarks())/3.0;

   fmark = test*0.2 + asgMarks*0.1 + eMark*0.7;
```

```
    return int(fmark);
}
};


 int main()
 {


    Marks marks;


    cout<< marks.getName() << ", " << marks.getNumber() << "has got "<<marks.calcMark() << "
marks " << endl;



    FinalMark myMark;


    cout<<myMark.getName() <<", "<< myMark.getNumber()<<"has final mark of
"<<myMark.calcMark() << " marks " << endl;



     return 0;
 }
```

f) The statement would be invalid because testMarks is not a public variable so only derived classes can access it.

Question 4

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


template<class T>

int count(vector<T> v1, T val)
{

   int j = 0;

   for ( int i = 0; i < v1.size(); i++)
      {
          if(val == v1[i])
          j++;
      }
         return j;
}




int main()
{
   vector<int> v1;
   vector<char> v2;

   char calpha, cval;
   int inum, ival;
```

```
cout << "Enter a list of numbers and end with number 0 " << endl;

cin >> inum;

while (inum!=0)
{
   v1.push_back(inum);
   cout << "Enter next number: " << endl;
   cin >> inum;
}

cout << "Enter number to search for: " << endl;

cin >> ival;

cout << ival << " occurs " << count<int>(v1, ival) << " times" << endl;

//
//

cout << "Enter alphabet letters and end with character # " << endl;

cin >> calpha;
calpha = tolower(calpha);

while (calpha!='#')
{
   v2.push_back(calpha);
   cout << "Enter next alphabet letter: " << endl;
   cin >> calpha;
```

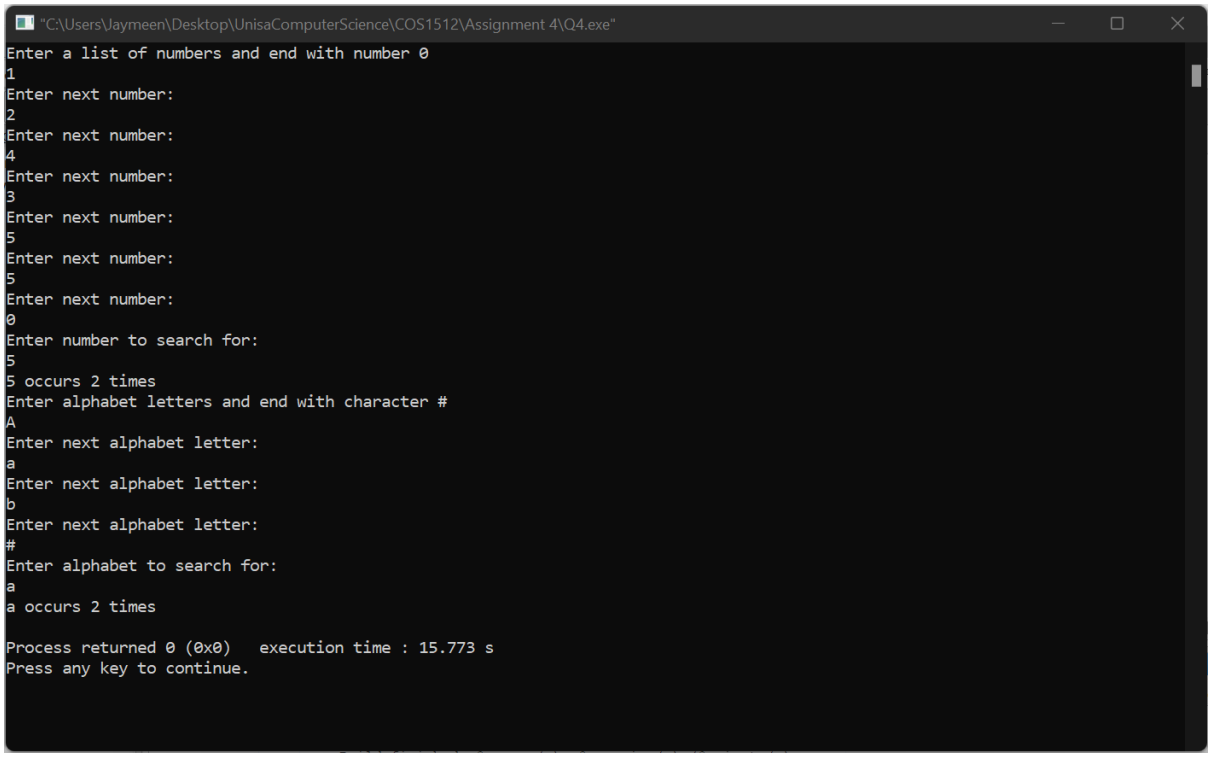
```
        calpha = tolower(calpha);
    }

    cout << "Enter alphabet to search for: " << endl;

    cin >> cval;
    cval = tolower(cval);

    cout << cval << " occurs " << count<char>(v2,cval) << " times" << endl;

    return 0;
}
```



Question 5

Main.cpp

```
//#include "Dictionary.cpp"
```

```
#include <iostream>
#include <cstdlib>
#include "Dictionary.h"
#include <vector>
#include <string>


using namespace std;


int main()
{
 Dictionary <int,string> parts();


 string part;

 int key;

 //add 4 values to the parts dictionary
 for (int i = 0; i <= 3; i++)
 {
 cout << "Please enter a part name and a key to add to the parts dictionary." << endl;

 cout << "Part name: ";

 getline(cin, part);

 cout << "Key for part name: ";

 cin >> key;
```

```
 parts.add(key, part);
 //cin.get();
 }

 cout << endl;

 parts.display();

 cout << endl;
 //find the part for a key

 cout << "For which key do you want to find the part? ";

 cin >> key;

 cout << "The part for key " << key << " is ";

 cout << parts.find(key) << endl;
 // cout << parts.find(100002);
 return 0;
}
```

Dictionary.cpp

```
#include "Dictionary.h"
#include <vector>
#include <iostream>
#include <cstdlib>
```

```
using namespace std;

template<class Tkeys, class Tvalues>
Dictionary<Tkeys, Tvalues>::Dictionary()
{
 //nothing to do, vector member variables are empty on
 //declaration
};

template<class Tkeys, class Tvalues>
void Dictionary<Tkeys, Tvalues>::add(Tkeys key,const Tvalues &value)
{
 keys.push_back(key);
 values.push_back(value);
}

template<class Tkeys, class Tvalues>
Tvalues Dictionary<Tkeys, Tvalues>::find (Tkeys key)
{
 for (int i = 0; i < keys.size(); i++)
 {
    if (key == keys[i])
    return values[i];
    else return "no such key can be found";
 }

}

template<class Tkeys, class Tvalues>
void Dictionary<Tkeys, Tvalues>::display()
```

```
{
 for (unsigned int i = 0; i < keys.size(); i++)
 cout << keys[i] << ' ' << values[i] << endl;

}
```

Dictionary.h

```
#ifndef DICTIONARY_H
#define DICTIONARY_H
#include <vector>
#include <string>
#include <iostream>
#include <cstdlib>


using namespace std;

template<class Tkeys, class Tvalues>

class Dictionary
{
 public:
 Dictionary();
 void add(Tkeys key,const Tvalues &value);
 Tvalues find (Tkeys key);
 void display();

 private:
 vector<Tkeys> keys;
 vector<Tvalues> values;
```

};

#endif // DICTIONARY_H

Cannot execute because of project error

Question 6