**INTI International College Penang School of Engineering and Technology**

**3+0 Bachelor of Science (Hons) in Computer Science, in collaboration with Coventry University, UK**

**3+0 Bachelor of Science (Hons) in Computing, in collaboration with Coventry University, UK**

**Coursework Cover Sheet**

|  |  |
| --- | --- |
| **Section A - To be completed by the student**  Full Name: Syamarvin A/L ravi(11324032)  Erik Yeong Zhi Hung(118430072) | |
| CU Student ID Number:Syamarvin (11324032) Erik (118430072) | |
| Semester: Aug 2022 | |
| Lecturer:  **Koo Lee Chun** | |
| Module Code and Title:  **5008CEM Programming for Developers** | |
| Assignment No. / Title:  **Courswork(Porfolio)** | % of Module Mark  **25%** |
| Hand out date:  **6 September 2022** | Due date:  **20 Nov 2022** |
| Penalties: No late work will be accepted. If you are unable to submit coursework on time due to extenuating circumstances you may be eligible for an extension. Please consult the lecturer. | |
| Declaration: I the undersigned confirm that I have read and agree to abide by the University regulations on plagiarism and cheating and Faculty coursework policies and procedures. I confirm that this piece of work is my own. I consent to appropriate storage of my work for plagiarism checking.  Signature(s): ---------------Hung------------Syamarvin----------- | |

**Programming Tasks**

**Question 1: Inventory Management System**

Write a program that implements a hash table to manage the inventory for a stationery store. The program shall fulfill the following requirements:

 Design a hash table to store a list of stationery inventory. Each stationery will has a stationery id (in string format, eg: S10001), description, quantity.

 Provide a menu that allows a user to perform the following operations until the option 4 (exit) is selected:

1) Stock in

2) Stock out

3) View summary report

4) Exit System

 If user selects option 1 (Stock in), the program shall ask user to enter the stationery id and quantity he/she wish to stock in. Increase the existing quantity accordingly in the hash table. Add the stationery information as new entry into the hash table if the stationery is not found.

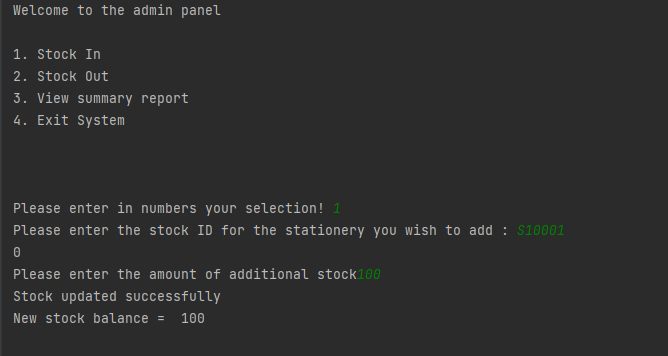
 If user selects option 2(stock out), the program shall ask user to enter the stationery id and quantity he/she wish to stock out. Reduce the existing quantity accordingly in the hash table. Prompt error to the user if fail to find the stationery to stock out.

 If user selects option 3 (View summary report), display a list of stationeries (including their quantity in tabular format) from the hash table.

 The program shall display appropriate messages or error messages when it is necessary.

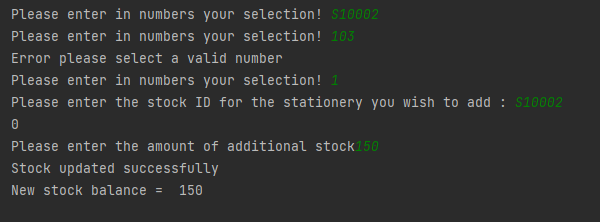
 The system shall demonstrate a good OOP design, data validation and error handling.

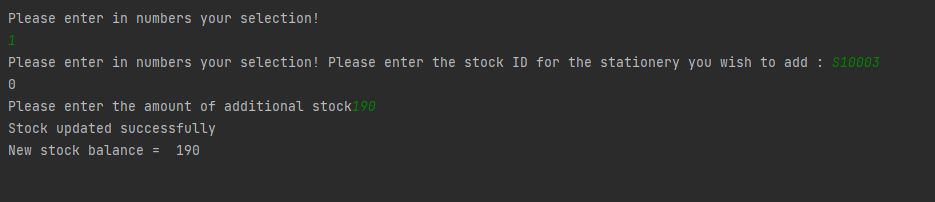
**CODE OUTPUT**



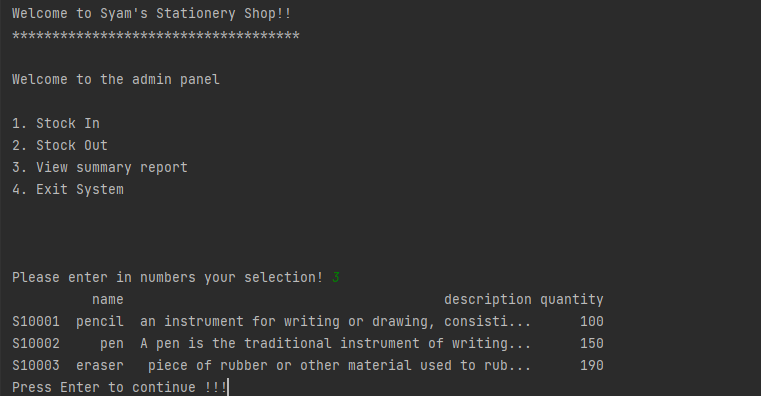
First instance of the program running where there are selecton for the users (Stock In,Stock Out,View Summary Report and Exit System.

Also demonstrates function for adding stock which is option **1.** The initial stock is 0 so we added 100 stock of pencils and it displays the new stock balance. Below will be the screen captures for adding stock for the remaining two items.

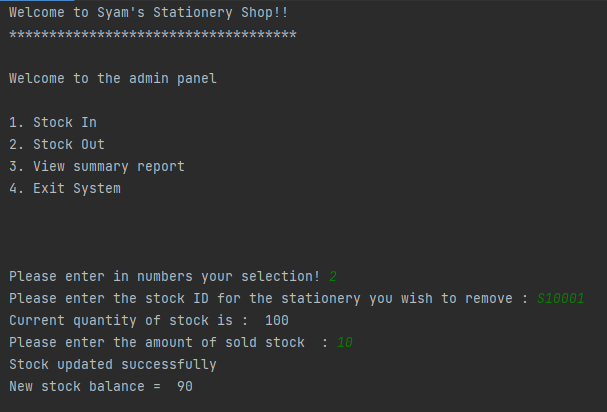


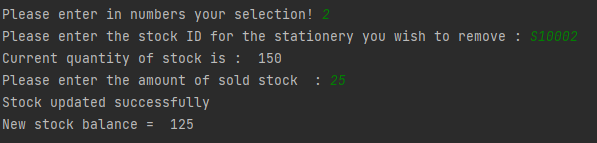
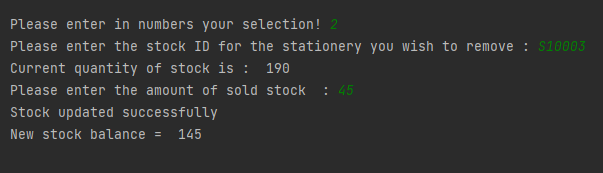


Add stock for pen and eraser

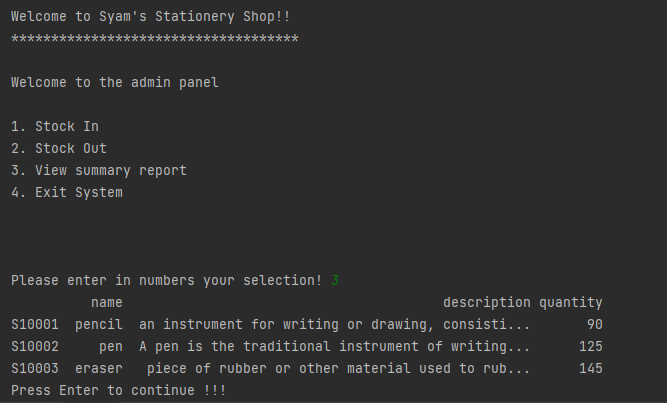


This output is from option **3** where it displays curent quantity of stock in hand. I jumped to the third option as to show first that user can view the stock they have added in from the previous snapshots above.

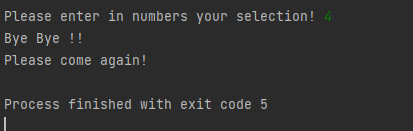


This shows the output for option **2** where user updates sold stocks for example, we added 100 pencils just now. And 10 has been sold therefore the program displays new stock balance as 90 (**100-10**). Same goes for the two more snapshot of the output below.

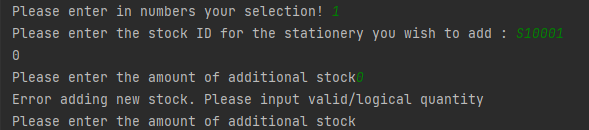
This shows the sale of the two other items (pen,eraser)



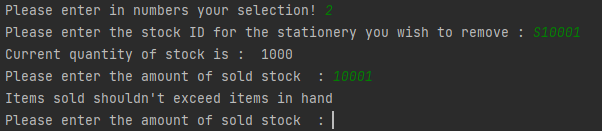
Here we jump pack to option 3 where we view the current amount of stock in hand after the sales.



Lastly option 4 where the system shuts down

**DATA VALIDATION**

If user were to add stock but entered a 0 or less it will throw an error to them asking them to input a logical quantity



If user try to declare more stock sold than whats in store the user will be shown an error saying that items sold should not exceed of that in stock

**System Weakness**

The weakness in this system is it is case sensitive for example the set stock id is **S10001** uf user were to insert **s10001** then the program will crash instantly and have to rerun it. Same goes to if user enter anything that is not the pre set stock id. Example if user enter **314jnf** the program will crash. I have tried to put a data validation to that but the program did not run at all therefore I have removed it entirely from the code.

**Reflection**

**Syamarvin** – Personally the question is not so hard as I have done similar things in C++ during diploma time and also with Java, therefore I suggested my teammate that we should do this in python as we can learn it in a different language for additional knowledge.

**Erik –** At first, I was confused with all the codes and methods. But then my teammate suggested we solve the question using python and Java, so I followed through with his suggestions and he has also taught a lot with Java and python.

**Appendix**

import pandas as pd  
import os  
  
stock = {  
 "S10001" : {"name": "pencil", "description" : "an instrument for writing or drawing, consisting of a thin stick of graphite or a similar substance enclosed in a long thin piece of wood or fixed in a cylindrical case." , "quantity" : 0},  
 "S10002" : {"name": "pen", "description" : "A pen is the traditional instrument of writing, so it is fitting that a writer be able to use words to create a vivid mental image of the tool of his craft" , "quantity" : 0},  
 "S10003" : {"name": "eraser", "description" : " piece of rubber or other material used to rub out marks made by ink, pencil, or chalk" , "quantity" : 0},  
}  
def stock\_in():  
 sID = str(input("Please enter the stock ID for the stationery you wish to add : "))  
 if sID in stock:  
 pass  
 else:  
 print("Stock ID provided is not found")  
 add\_stock()  
  
 ori\_quantity = stock[sID]["quantity"]  
 print(ori\_quantity)  
 while True:  
 try:  
 new\_stock = int(input("Please enter the amount of additional stock"))  
 if new\_stock > 0:  
 new\_quantity = ori\_quantity + new\_stock  
 stock[sID]["quantity"] = new\_quantity  
 print("Stock updated successfully")  
 print("New stock balance = ",stock[sID]["quantity"])  
 break  
 else :  
 print("Error adding new stock. Please input valid/logical quantity")  
 except:  
 pass  
 menu()  
  
   
  
def stock\_out():  
 sID = str(input("Please enter the stock ID for the stationery you wish to remove : "))  
 if sID in stock:  
 pass  
 else:  
 print("Stock ID provided is not found")  
 add\_stock()  
  
 ori\_quantity = stock[sID]["quantity"]  
 print("Current quantity of stock is : " , ori\_quantity)  
 while True:  
 try:  
 new\_stock = int(input("Please enter the amount of sold stock : "))  
 if new\_stock > 0 and new\_stock < ori\_quantity:  
 new\_quantity = ori\_quantity - new\_stock  
 stock[sID]["quantity"] = new\_quantity  
 print("Stock updated successfully")  
 print("New stock balance = ",stock[sID]["quantity"])  
 break  
 else :  
 print("Items sold shouldn't exceed items in hand")  
 except:  
 pass  
 menu()  
  
  
  
def summary\_report():  
 df = pd.DataFrame(stock).T  
 df.fillna(0, inplace=True)  
 print(df)  
 input("Press Enter to continue !!!")  
 menu()  
   
  
def exit\_menu():  
 print("Bye Bye !! ")  
 print("Please come again!")  
 os.\_exit(5)  
  
   
def case(selection):  
 if selection == 1 :  
 stock\_in()  
 elif selection == 2 :  
 stock\_out()  
 elif selection == 3:  
 summary\_report()  
 elif selection == 4:  
 exit\_menu()  
 else :  
 menu()  
  
   
  
   
def menu():  
 print("")  
 print("")  
 print("Welcome to Syam's Stationery Shop!!")  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("")  
 print("Welcome to the admin panel")  
 print("")  
 print("1. Stock In ")  
 print("2. Stock Out")  
 print("3. View summary report")  
 print("4. Exit System")  
 print("")  
 print("")  
 print("")  
   
 while True:  
 try:  
 selection = int(input("Please enter in numbers your selection! "))  
 if selection > 0 and selection < 5:  
 break  
 else :  
 print("Error please select a valid number")  
 continue  
 except:  
 pass  
  
 case(selection)   
   
menu()

The source Code for Question 1

**Question 2: Binary Search Trees Run Time**

Write a program to compare the runtime taken to search a number from different the binary search tree structures. The program must fulfill the following requirements:

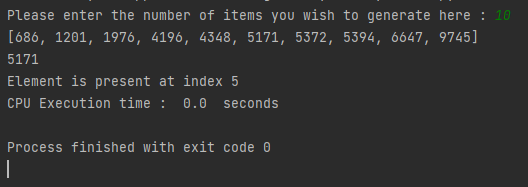
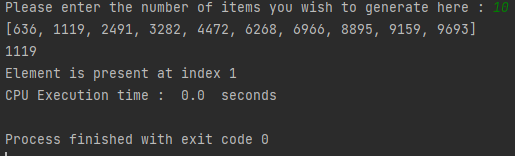
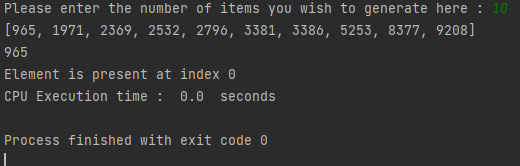
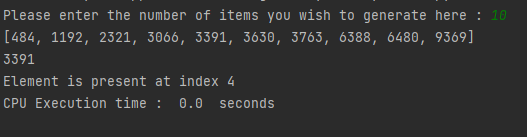
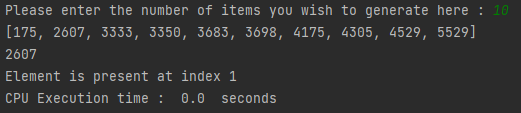
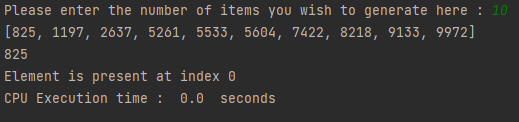
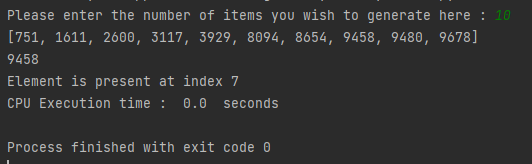
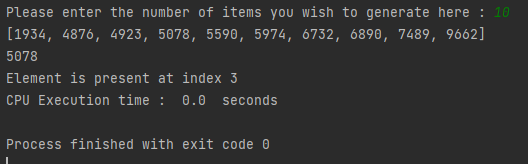
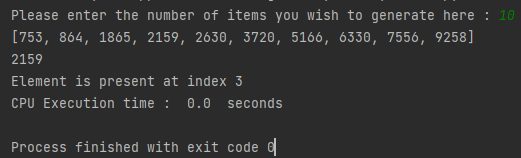
 Generate n numbers randomly (the random numbers shall between 1-10000, no duplicate). The n value shall be determined by the user.

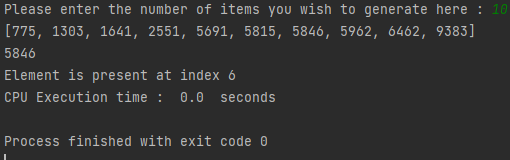
 Use the generated ramdom number to prepare two datasets: random order list and a sorted list. Then insert those data into two separate Binary search tree.

 Run the program 10 times (with n value of 10) and calculate the minimum, maximum and avarage time taken to search a number 20000 (a number that never exist in any of BST).

 Repeat this test for n value of 100, 1000, 10,000.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Observe the output and complete the following table. Discuss the theorical concept of BST in term of run time to support your observation. n | | | Total time taken | | |
| Sorted | | | Unsorted (random list) | | |
| Minimum | Maximum | Average | Minimum | Maximum | Average |
| 10 | | | | | |
| 100 | | | | | |
| 1000 | | | | | |
| 10000 | | | | | |



Output when the code is run 10 times with all cpu execution time of 0.0 seconds

**System Weakness**

The system cannot display unsorted list. And so far the average cpu run time is 0

**Reflection**

Syamarvin – This question is the one I was stuck with most as I could not find a way yet to display the unsorted list without compromising the entire question

Erik – After finishing with my part of the assignment, I tried lending a helping hand with solving question 2 as my teammate was having trouble solving it by himself. But I as well was stuck on this question particularly unable to code the system into displaying an unsorted list.

**Appendix**

import random  
import time  
  
  
def sorted\_numlist(n):  
 sortedlist = []  
 sortedlist = random.sample(range(1, 10001), n)  
 sortedlist.sort()  
 return sortedlist  
  
  
def random\_numlist(n):  
 randomlist = []  
 randomlist = random.sample(range(1, 10001), n)  
 return randomlist  
  
  
def binary\_search(arr, low, high, x):  
 if high >= low:  
  
 mid = (high + low) // 2  
 if arr[mid] == x:  
 return mid  
 elif arr[mid] > x:  
 return binary\_search(arr, low, mid - 1, x)  
 else:  
 return binary\_search(arr, mid + 1, high, x)  
 else:  
 return -1  
  
  
start\_time = time.process\_time()  
  
n = int(input("Please enter the number of items you wish to generate here : "))  
x = sorted\_numlist(n)  
print(x)  
search = random.choice(x)  
print(search)  
first = 0  
result = binary\_search(x, 0, len(x) - 1, search)  
  
if result != -1:  
 print("Element is present at index", str(result))  
else:  
 print("Element is not present in array")  
  
end\_time = time.process\_time()  
exec\_time = end\_time - start\_time  
print('CPU Execution time : ', exec\_time, ' seconds')

Code For Question 2

**Question 3: Graph**

Write a program that uses graph concept. The program shall fulfil the following requirements:

 construct a class graph that consists of the following methods:

o addEdges : add edges with weight to a graph

o countAdjacentVertex : calculate and return number of adjacent vertex (neightbor)

o countOutDegree : calculate and return total outdegree for a given vertex

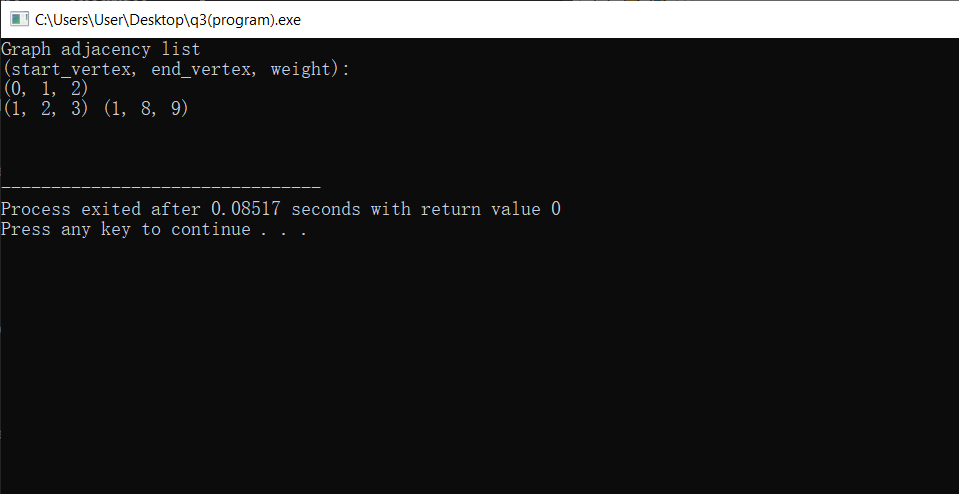
o countInDegree : calculate and return total indegee for a given vertex

o isDirectedGraph : return true if the graph is directed graph, otherwise, return false

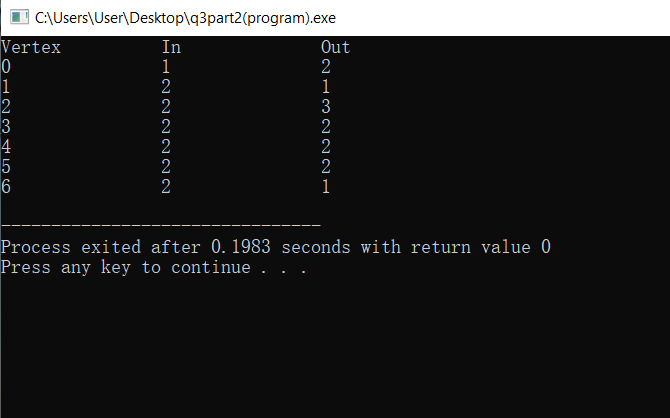
 Create two graph objects to present a directed and indirected and test all the above functions.

**System Weakness:**

The weakness of the solution given for this question is that the program doesn’t include the method of converting the graph into directed and in-directed.



The above diagram shows that the adding of edges to a vertex and calculating and returning the number of adjacent vertex.



The above diagram shows the program calculated and returned the outdegree and indrgree of the vertex given.

**Appendix**

(code for question 3 for addEdges and countAdjacentVertex)

#include <iostream>

using namespace std;

struct adjNode {

int val;

int cost;

adjNode\* next;

};

struct graphEdge {

int start\_ver;

int end\_ver;

int weight;

};

class Graph{

adjNode\* getAdjListNode(int value, int weight, adjNode\* head) {

adjNode\* newNode = new adjNode;

newNode->val = value;

newNode->cost = weight;

newNode->next = head;

return newNode;

}

int N;

public:

adjNode \*\*head;

Graph(graphEdge edges[], int n, int N) {

head = new adjNode\*[N]();

this->N = N;

for (unsigned i = 0; i < n; i++) {

int start\_ver = edges[i].start\_ver;

int end\_ver = edges[i].end\_ver;

int weight = edges[i].weight;

adjNode\* newNode = getAdjListNode(end\_ver, weight, head[start\_ver]);

head[start\_ver] = newNode;

}

}

};

void display\_AdjList(adjNode\* ptr, int i)

{

while (ptr != nullptr) {

cout << "(" << i << ", " << ptr->val

<< ", " << ptr->cost << ") ";

ptr = ptr->next;

}

cout << endl;

}

int main()

{

graphEdge edges[] = {

{1,2,3},{4,5,6},{7,8,9},{0,1,2},

};

int N = 4; // Number of vertices in the graph

int n = sizeof(edges)/sizeof(edges[0]);

Graph graph(edges, n, N);

cout<<"Graph\_Adjacency\_List "<<endl<<"(start\_vertex\t\t end\_vertex\t\t weight):"<<endl;

for (int i = 0; i < N; i++)

{

display\_AdjList(graph.head[i], i);

}

return 0;

}

(code for question 3 for countOutDegree and countInDegree)

#include <bits/stdc++.h>

using namespace std;

void findInOutDegree(vector<vector<int>> AdjList,

int n)

{

vector<int> iN(n,0);

vector<int> ouT(n,0);

for(int i=0;i<n;i++)

{

// Out degree for ith vertex will be the count

// of direct paths from i to other vertices

ouT[i] = AdjList[i].size();

for(int j=0;j<AdjList[i].size();j++)

iN[AdjList[i][j]]++;

}

cout << "Vertex\t\tIn\t\tOut" << endl;

for(int k = 0; k < n; k++)

{

cout << k << "\t\t"

<< iN[k] << "\t\t"

<< ouT[k] << endl;

}

}

int main()

{

vector<vector<int>> AdjList;

vector<int> tmp;

tmp.push\_back(1);

tmp.push\_back(2);

AdjList.push\_back(tmp);

tmp.clear();

tmp.push\_back(3);

AdjList.push\_back(tmp);

tmp.clear();

tmp.push\_back(0);

tmp.push\_back(5);

tmp.push\_back(6);

AdjList.push\_back(tmp);

tmp.clear();

tmp.push\_back(1);

tmp.push\_back(4);

AdjList.push\_back(tmp);

tmp.clear();

tmp.push\_back(2);

tmp.push\_back(3);

AdjList.push\_back(tmp);

tmp.clear();

tmp.push\_back(4);

tmp.push\_back(6);

AdjList.push\_back(tmp);

tmp.clear();

tmp.push\_back(5);

AdjList.push\_back(tmp);

tmp.clear();

int n = AdjList.size();

findInOutDegree(AdjList, n);

}

**Question 4: Concurrent process**

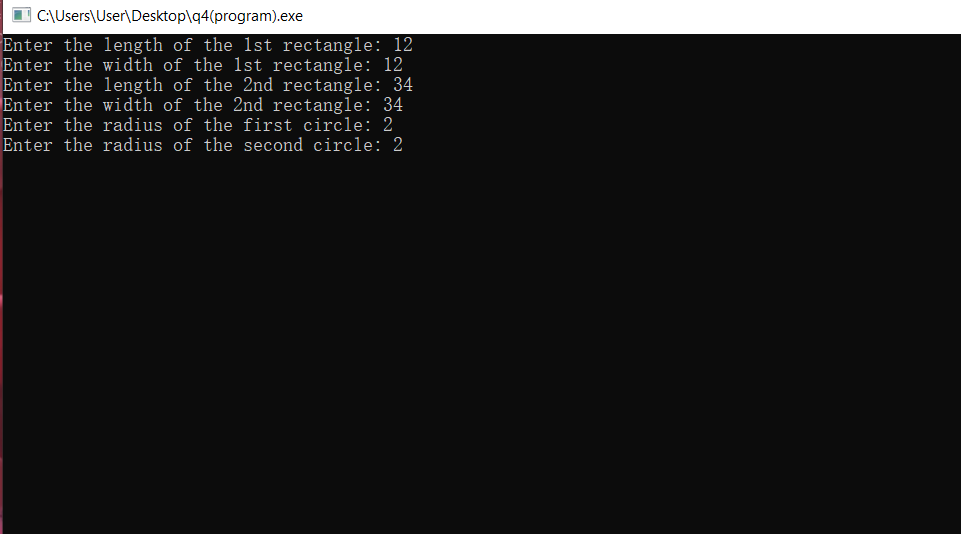
Define two functions: calc\_areaOfRectangle and calc\_areaOfCircle that receive appropriate parameters. The function shall calculate and display the area of rectangle and circle respectively.

Write a program that asks user to enter length and width of two rectangles, and radius of two circles. Then, the program shall call the appropriate functions concurrently to calculate and display the areas of those shapes.

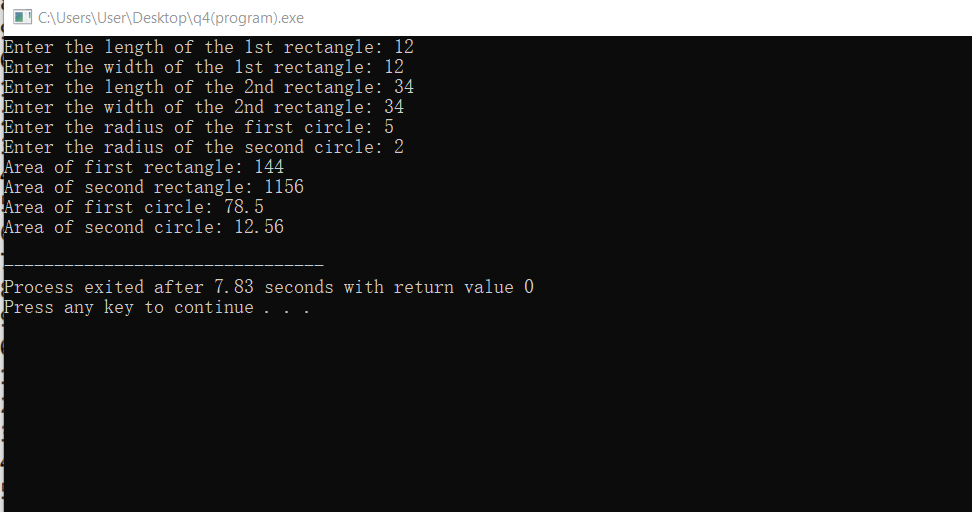
Observe the output of your solution. Discuss the theorical behind to support your observation in the documentation.

**System Weakness:**

The solution I gave for this question is not perfect in that I didn’t use the method of multithreading in the coding phase. I have tried to implement multithreading and I was unsuccessful.



In the above diagram, the users input the lengths and widths for the 2 rectangles as well as the radius for the 2 circles in order for the system to do the calculations



The above diagram shows the result of areas of the rectangles and circles after inputting all the required lengths, width and radius of the shapes.

**Appendix**

(code for question 4)

#include <iostream>

#include <cmath>

using namespace std;

double calculate\_areaOfRectangle(double length, double width)

{

double area=length\*width;

}

float calculate\_areaOfCircle(float radius)

{

float pi=3.14;

float area=pi\*radius\*radius;

}

int main()

{

double length1;

double width1;

double length2;

double width2;

float radius1;

float radius2;

cout<<"Enter the length of the 1st rectangle: ";

cin>>length1;

cout<<"Enter the width of the 1st rectangle: ";

cin>>width1;

cout<<"Enter the length of the 2nd rectangle: ";

cin>>length2;

cout<<"Enter the width of the 2nd rectangle: ";

cin>>width2;

cout<<"Enter the radius of the first circle: ";

cin>>radius1;

cout<<"Enter the radius of the second circle: ";

cin>>radius2;

cout<<"Area of first rectangle: "<<calculate\_areaOfRectangle(length1,width1)<<endl;

cout<<"Area of second rectangle: "<<calculate\_areaOfRectangle(length2,width2)<<endl;

cout<<"Area of first circle: "<<calculate\_areaOfCircle(radius1)<<endl;

cout<<"Area of second circle: "<<calculate\_areaOfCircle(radius2)<<endl;

}

**Reflection**

Reflection (Erik Yeong Zhi Hung)

Throughout my time in trying to solve the questions given, specifically question 3, I was stuck on the question ever since I finish question 4 as I don’t have an expansive knowledge on the codes and knowhow to construct a graph consisting of multiple methods in C++. And now, I can safely say that not only have I completed the question, and that I also have understood the basic concept of constructing a graph using C++ and will be using it for my future IT career.

**Reference list**

By: ShrabanaBiswas Written:  18 Feb, 2022

<https://www.geeksforgeeks.org/add-and-remove-edge-in-adjacency-list-representation-of-a-graph/>

By: Janice-shah, Written: 13 Sep, 2022

<https://www.geeksforgeeks.org/finding-in-and-out-degrees-of-all-vertices-in-a-graph/>