

UNIVERSITY COLLEGE OF TECHNOLOGY SARAWAK (UCTS)

SCHOOL OF COMPUTING

YEAR 1, SEMESTER 2, SESSION 2018

**CSS 3133 Object Oriented Programming**

**Object-Oriented Programming Assignment**

PREPARED FOR:

S.M. Anli Sherine

Group Name : Group zero

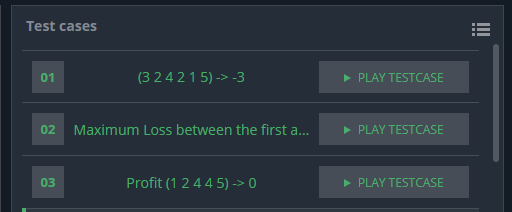
Group Leader : Then Vui Lin

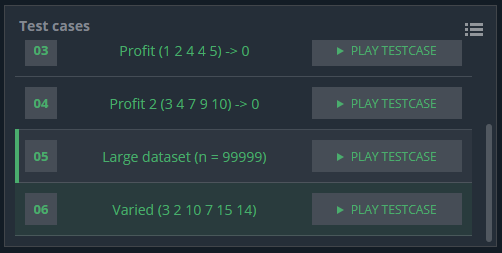
PREPARED BY:

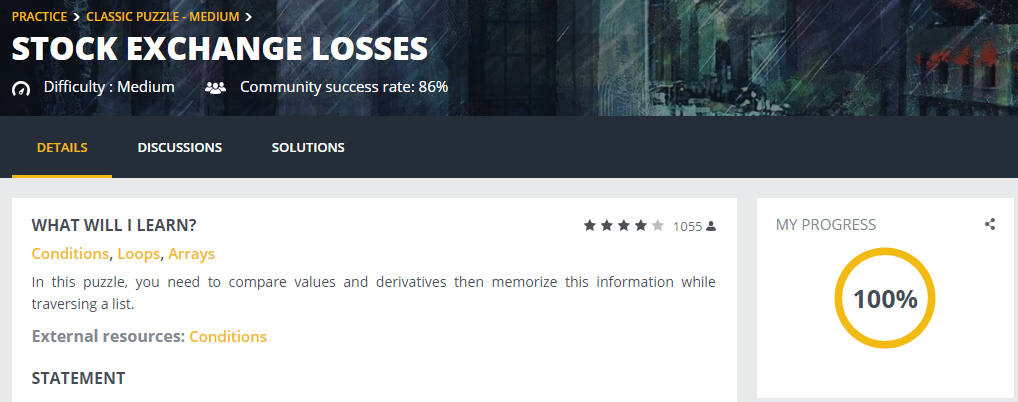
|  |  |
| --- | --- |
| Then Vui Lin | BCS 1704 0003 |
| Chen Zhi Hao | BCS 1704 0004 |
| Sia Yuk Heng | BCS 1704 0001 |
| Daniel Wong Tze Yiu | BCS 1707 0005 |

**Training Modules**

**Achieved Test Cases for Stock Exchange Losses:**







**Program flowchart for Stock Exchange Losses:**

True

Read v[i]

i = 0

Declare integer array v[n]

Read n

Declare integer variables n, i, loss, maxindex, minindex

Start

False

i < n?

i+1

A

Declare integer variables max and min   
max = v[0]  
min = v[0]

A

C

B

Declare integer variable temp  
temp = max  
max = min  
min = temp  
Declare integer variable loss2  
loss2 =0

False

True

minindex < maxindex?  
 and  
maxindex != n-1?

True

False

True

True

False

False

v[i] < min?

v[i] > max?

i+1

i < n?

End

Write  
-loss

v[i] = min

minindex = i

loss = max - min

max = v[i]

maxindex = i

i =1

B

False

False

True

loss = loss2

loss = loss

i+1

max = v[i]

maxindex = i

True

v[i] = min

minindex = i

loss2 = max - min

i = minindex+1

i < n?

C

True

v[i] > max?

False

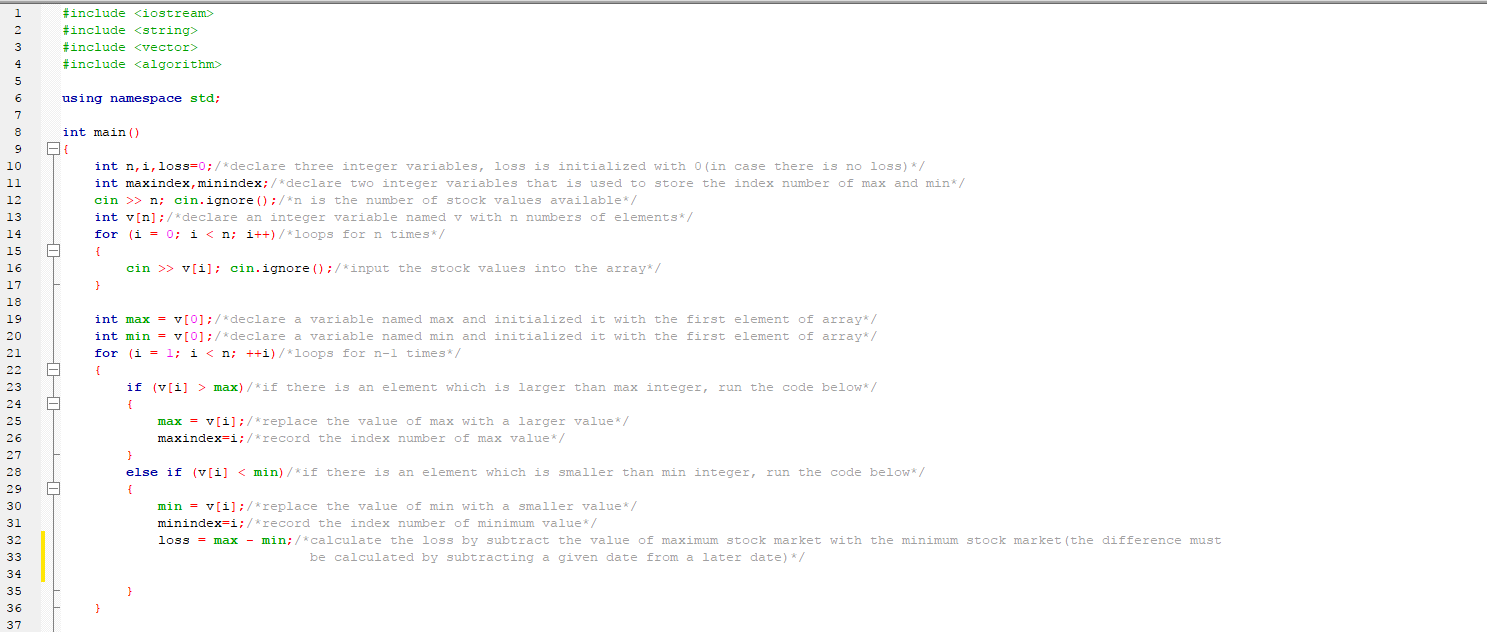
True

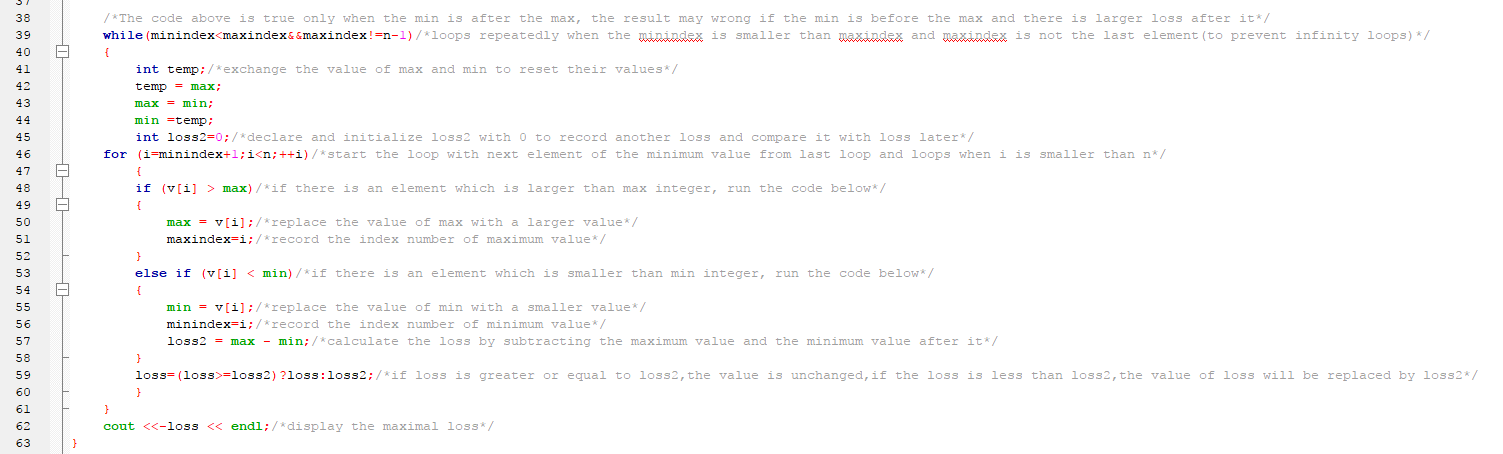
v[i] < min?

False

loss >= loss2?

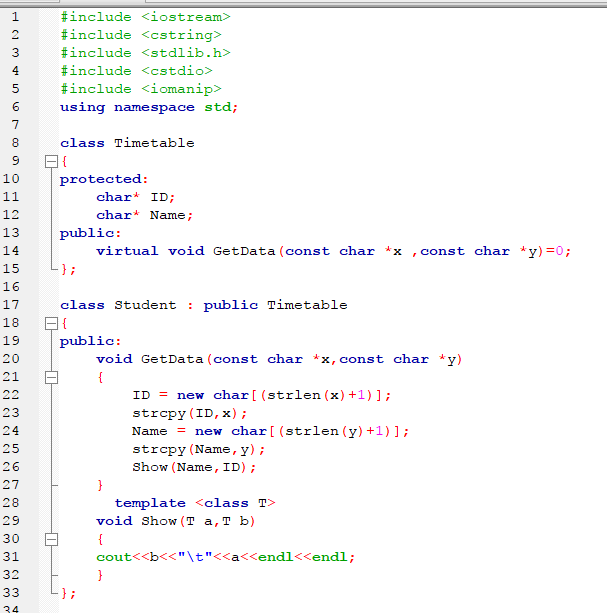
**Program Codes for Stock Exchange Losses(including inline comments) :**

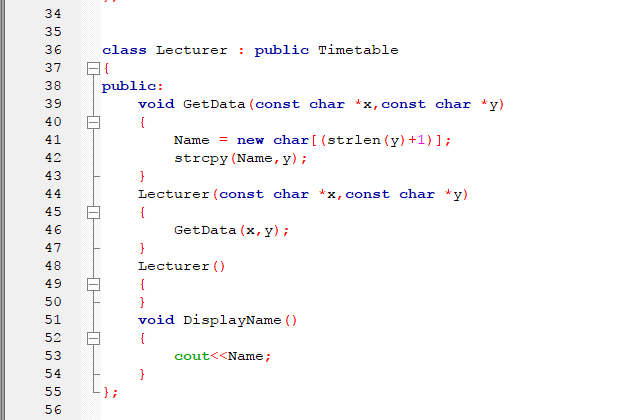


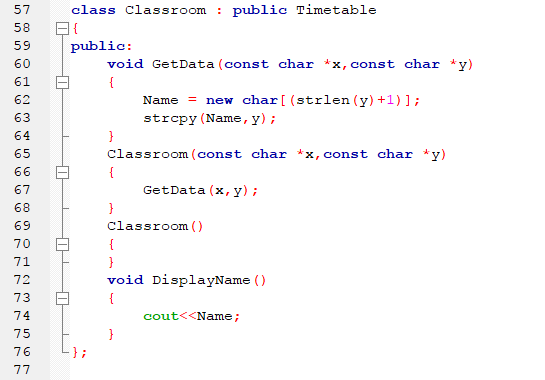


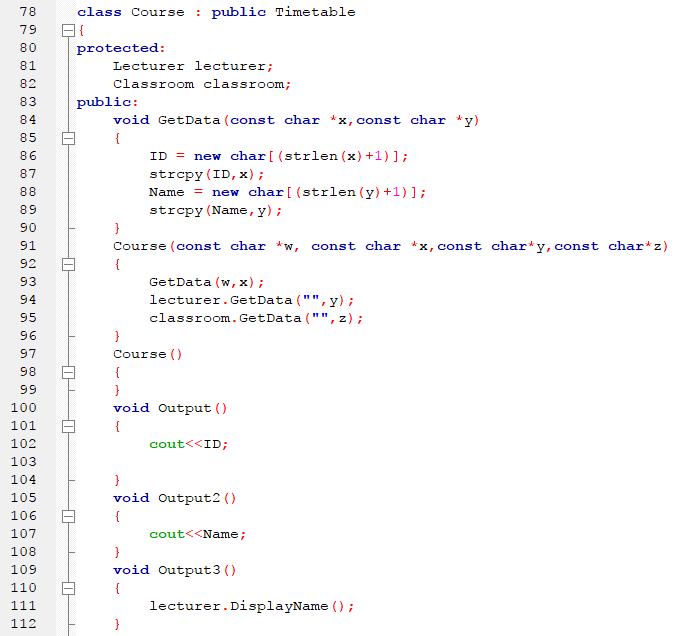
**Student Course Timetable Arrangement System**

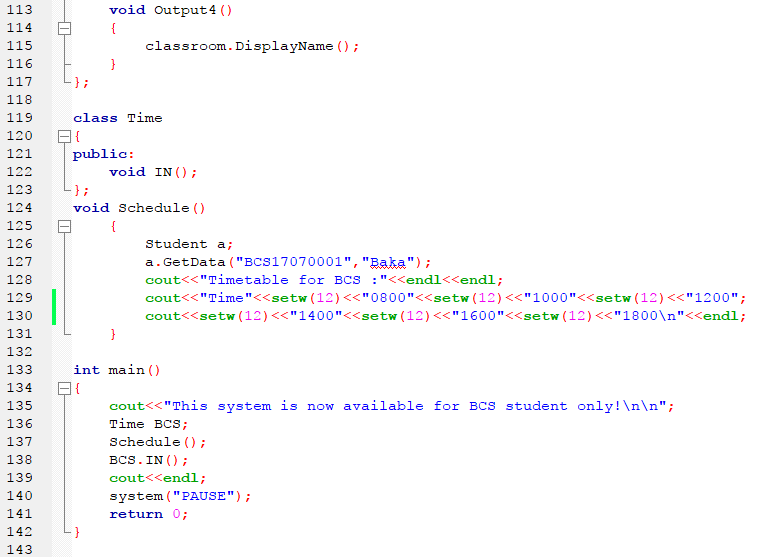
**Program Code:**













**Flowchart:**

Schedule()

Creating object of class Time named BCS.

write “This system is now available for BCS student only!”

Start

BCS->void Time::IN()

End

Schedule()

The values of arguments, x and y are copied to ID and Name.

void Student::GetData(const char \*x,const char \*y)

a-> void Student::GetData("BCS17070001","Baka")

End

Write b a

Show(T a,T b)

End

Show(Name,ID)

End

Write “Time 0800 1000 1200 1400 1600 1800”

write “Timetable for BCS:”

Creating object of class Student named a

subj1 create an object of class Lecturer named lecturer

Create an object of class Course named subj1

void Time::IN()

A

subj4 create an object of class Classroom named classroom

subj4 create an object of class Lecturer named lecturer

Create an object of class Course named subj4

subj3->Course::Course("CSS3533","OS","Gary","R923")

subj3 create an object of class Classroom named classroom

subj3 create an object of class Lecturer named lecturer

Create an object of class Course named subj3

subj2->Course::Course("CSS3313","Multimedia","Yiiong","Lab3")

subj2 create an object of class Classroom named classroom

subj2 create an object of class Lecturer named lecturer

Create an object of class Course named subj2

subj1->Course::Course("CSS3542","UNIX","Hidayu","Lab3")

subj1 create an object of class Classroom named classroom

subj4->Course::Course("CSS3323","Database","Fadli","R223")

A

B

subj7->Course::Course("CSS3223","Calculus","Yiiong","R923")

subj7 create an object of class Classroom named classroom

subj7 create an object of class Lecturer named lecturer

Create an object of class Course named subj7

subj6->Course::Course("CSS3133","OOP","Anli","Lab4")

subj6 create an object of class Classroom named classroom

subj6 create an object of class Lecturer named lecturer

Create an object of class Course named subj6

subj5->Course::Course("CSS3122","Data Structure","Chew Kim Mey","Lab3")

subj5 create an object of class Classroom named classroom

subj5 create an object of class Lecturer named lecturer

Create an object of class Course named subj5

subj2->Course::Output()

subj1->Course::Output()

write “Mon”

B

write   
“Wed”

C

subj3->Course::Output4()

subj4->Course::Output3()

subj3->Course::Output3()

subj4->Course::Output2()

subj3->Course::Output2()

subj4->Course::Output()

subj3->Course::Output()

subj5->Course::Output4()

subj4->Course::Output4()

subj5->Course::Output3()

subj4->Course::Output3()

subj5->Course::Output2()

subj4->Course::Output2()

subj5->Course::Output()

subj4->Course::Output()

write “Tue”

subj2->Course::Output4()

subj1->Course::Output4()

subj2->Course::Output3()

subj1->Course::Output3()

subj2->Course::Output2()

subj1->Course::Output2()

subj6->Course::Output()

write   
“Thur”

subj4->Course::Output4()

C

End

subj2->Course::Output4()

subj7->Course::Output4()

subj2->Course::Output3()

subj7->Course::Output3()

subj2->Course::Output2()

subj7->Course::Output2()

subj2->Course::Output()

subj7->Course::Output()

write   
“Fri”

subj7->Course::Output4()

subj6->Course::Output4()

subj7->Course::Output3()

subj6->Course::Output3()

subj7->Course::Output2()

subj6->Course::Output2()

subj7->Course::Output()

Course::Course(const char \*w, const char \*x,const char\*y,const char\*z)

End

The value of argument y is copied to Name.

Classroom::GetData(const char \*x,const char \*y)

End

The value of argument y is copied to Name.

Lecturer::GetData(const char \*x,const char \*y)

End

The values of arguments, x and y are copied to ID and Name.

Course::GetData(const char \*x,const char \*y)

End

classroom->Classroom::GetData(“”,”z”)

lecturer->Lecturer::GetData(“”,”y”)

Course::GetData(w,x)

End

Write ID

void Course::Output()

End

Write Name

void Classroom::DisplayName()

End

Write Name

void Lecturer::DisplayName()

End

classroom->Classroom::DisplayName()

void Course::Output4()

End

lecturer->Lecturer::DisplayName()

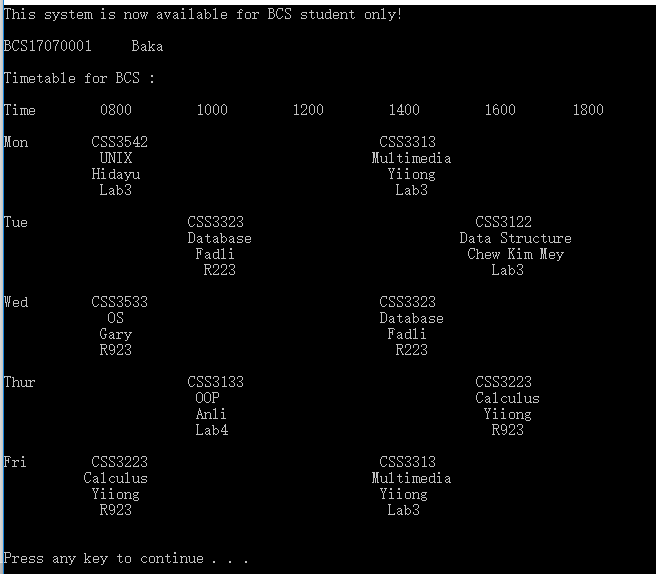
void Course::Output3()

End

Write Name

void Course::Output2()

**Output:**



**Github:**

1. Sia Yuk Heng BCS17040001

<https://github.com/re0saya/test>

2. Then Vui Lin BCS17040003

<https://github.com/sinmicsin/OOP-Assignment>

3. Chen Zhi Hao BCS17040004

<https://github.com/Emodarryl/Emohao>

4. Daniel Wong Tze Yiu

<https://github.com/ahnielwong99/OOP-Assignment>