TUNKU ABDUL RAHMAN UNIVERSITY COLLEGE

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

(Bachelor of Science (Honours) in Management Mathematics With Computing)

Year 2 Semester 3

RMM (G1)

Practical Assignment

BACS2093 Operating Systems (AY 202101)

Name(Block Capital)	Registration No	Signature	Marks
1. HEE SZE WEI	19WMR05920		
2. LEE SHU ERN	19WMR05933		

Lecturer/Tutor's Name: Ms Chin Chai Lim

Task 1:

Screen Output (s)

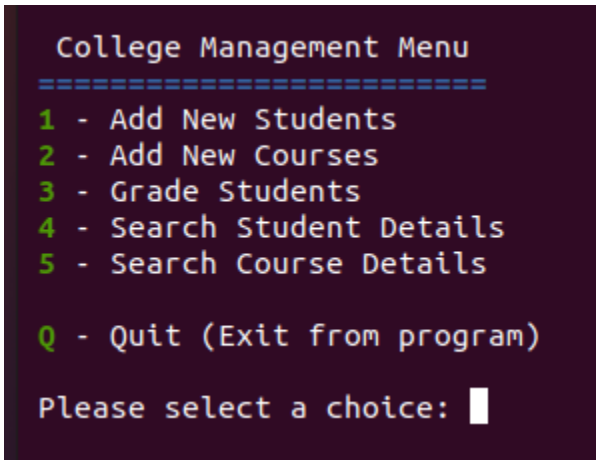


Figure 1.1 College Management Menu

Users are required to choose one of the following choices.

Input Validation

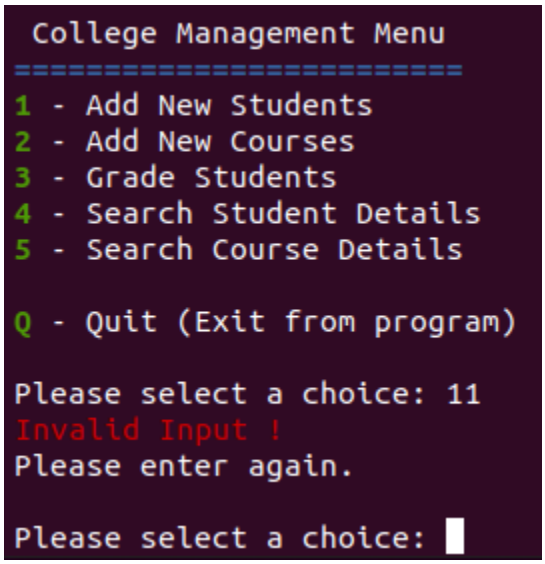


Figure 1.2

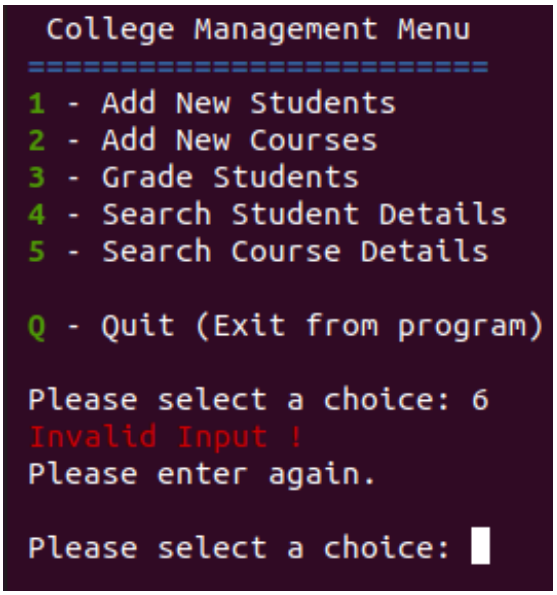


Figure1.3

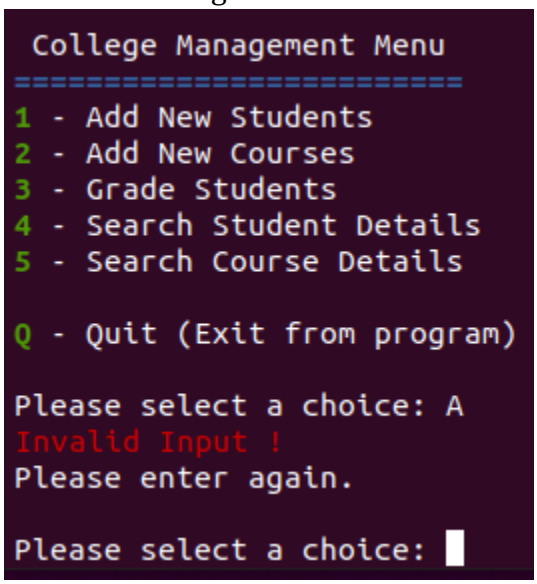


Figure 1.4

Figure 1.2 - 1.4 show the input validation. Users can only enter 1-5 and Q. If the user inputs numbers other than 1 to 5 and alphabet other than Q, the user needs to re-enter her/his choice.

Sample Code
College Management Menu (ManagementMenu)

```
#!/bin/bash

if ! [ -f $course.txt ]
then touch course.txt
fi

if ! [ -f $student.txt ]
then touch student.txt
fi

echo " "
echo "  College Management Menu"
echo -e "\e[1;34m ======================== \e[0m"  #blue text
echo -e "\e[1;32m 1 \e[0m- Add New Students " #Green
echo -e "\e[1;32m 2 \e[0m- Add New Courses "
echo -e "\e[1;32m 3 \e[0m- Grade Students "
echo -e "\e[1;32m 4 \e[0m- Search Student Details "
echo -e "\e[1;32m 5 \e[0m- Search Course Details "
echo " "
echo -e "\e[1;32m Q \e[0m- Quit (Exit from program) "
echo " "

validChoice=false
while [ $validChoice = "false" ]
do
 echo -n " Please select a choice: "; read choice
 if ! [[ $choice =~ ^[Qq1-5]$ ]]
  then
    echo -e "\033[31;5m Invalid Input ! \033[0m" #blink red text
    echo " Please enter again."
    echo " "
    validChoice=false
 else
    validChoice=true
  fi
done

 case $choice in
  1) echo " Displaying Programme Selections Menu..."
     sleep 2
     echo " "
    ./prog
    ;;

  2) echo " Displaying Add New Courses Form..."
     sleep 2
     echo " "
     ./addCourse
     ;;

  3) echo " Displaying Student Validation Form..."
     sleep 2
     echo " "
     ./studV
     ;;

  4) echo " Displaying Search Student Details Form..."
     sleep 2
     echo " "
     ./searchStud
     ;;
```

```
62
63   5) echo " Displaying Search Course Details Form..."
64      sleep 2
65      echo " "
66      ./searchCourse
67      ;;
68
69   [Qq]) echo " Closing the program..."
70         sleep 2
71         echo -e "\e[1;33m Bye Bye ! \e[0m" #yellow
72         exit
73         ;;
74  esac
```

Task 2:

Screen Output(s)

PART A: When user select to add new student

```
Please select a choice: 1
Displaying Programme Selections Menu...

 Programme Selections Menu
============================
T - RIT ( Bachelor in Information Technology )
D - RSD ( Bachelor in Software Development )
S - RST ( Bachelor in Interactive Software Technology )
E - REI ( Bachelor in Enterprise Information System )
F - RSF ( Bachelor in Software Engineering )
Q - Quit ( Return to College Management Menu)

Please select a choice:
```

Figure 2.1 Programme Selections Menu

If the user chooses 1 from the previous menu, Programme Selections Menu will be displayed.

```
 Programme Selections Menu
============================
T - RIT ( Bachelor in Information Technology )
D - RSD ( Bachelor in Software Development )
S - RST ( Bachelor in Interactive Software Technology )
E - REI ( Bachelor in Enterprise Information System )
F - RSF ( Bachelor in Software Engineering )
Q - Quit ( Return to College Management Menu)

Please select a choice: A
Invalid Input !
Please enter agian.
Please select a choice:
```

Figure 2.2 Input Validation

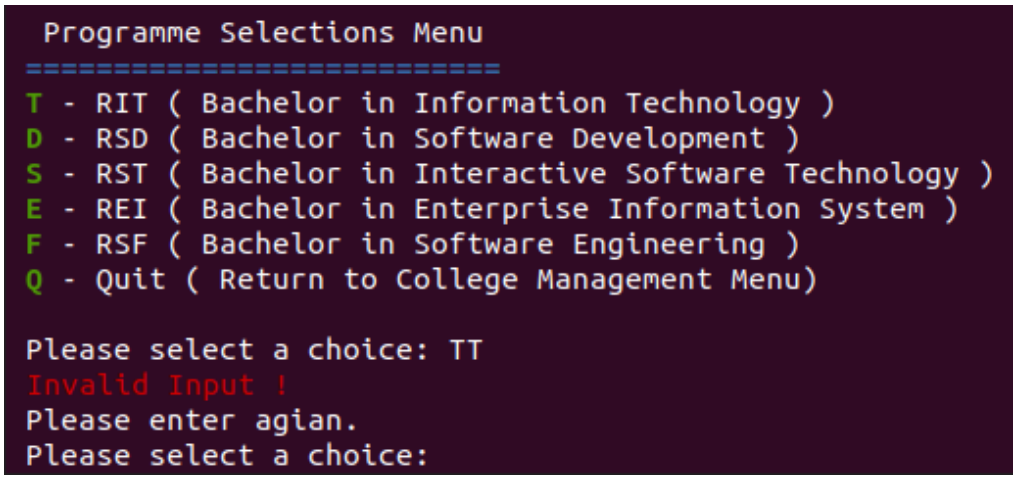


Figure 2.3 Input Validation

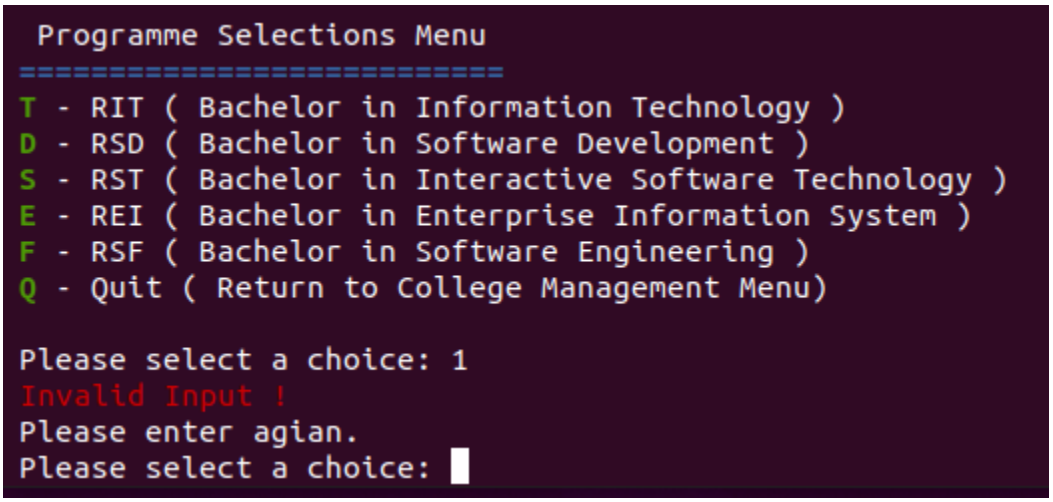


Figure 2.4 Input Validation

Figure 2.2 and 2.3 show the input validation for programme selections menu. If the user inputs an alphabet other than T, D, S, E, F, Q and numbers, it will prompt a message “Invalid Input”. Then, users are required to enter the selection again.

```
 Programme Selections Menu
============================
T - RIT ( Bachelor in Information Technology )
D - RSD ( Bachelor in Software Development )
S - RST ( Bachelor in Interactive Software Technology )
E - REI ( Bachelor in Enterprise Information System )
F - RSF ( Bachelor in Software Engineering )
Q - Quit ( Return to College Management Menu)

Please select a choice: T
Programme Selected:  RIT ( Bachelor in Information Technology )
User is requested to fill in the Student Details Form.
```

Figure 2.5
After selecting the programme, the user is requested to fill up the Student Details Form.

```
 Programme Selections Menu
============================
T - RIT ( Bachelor in Information Technology )
D - RSD ( Bachelor in Software Development )
S - RST ( Bachelor in Interactive Software Technology )
E - REI ( Bachelor in Enterprise Information System )
F - RSF ( Bachelor in Software Engineering )
Q - Quit ( Return to College Management Menu)

Please select a choice: T
Programme Selected:  RIT ( Bachelor in Information Technology )
User is requested to fill in the Student Details Form.

 Add New Students Form
------------------------
Student ID (00AAA00000):19WMR05920
Student Name           :Hee Sze Wei
Birth Date (YYYY-MM-DD):2000-02-16
Contact Number         :016-6918841
Mailing Address        :2170, Taman Permai 3, 70200 Seremban
Email Address          :szewei@tarc.edu.my
Saving...

Add Another Student? (y)es or (q)uit :
```

Figure 2.6 Add New Student Form
After the user has selected a programme, the user is required to fill up the student details such as Student ID, Student Name, Birth Date, Contact Number, Mailing Address and Email Address. After the user filled up the form, the information will be saved to student.txt.

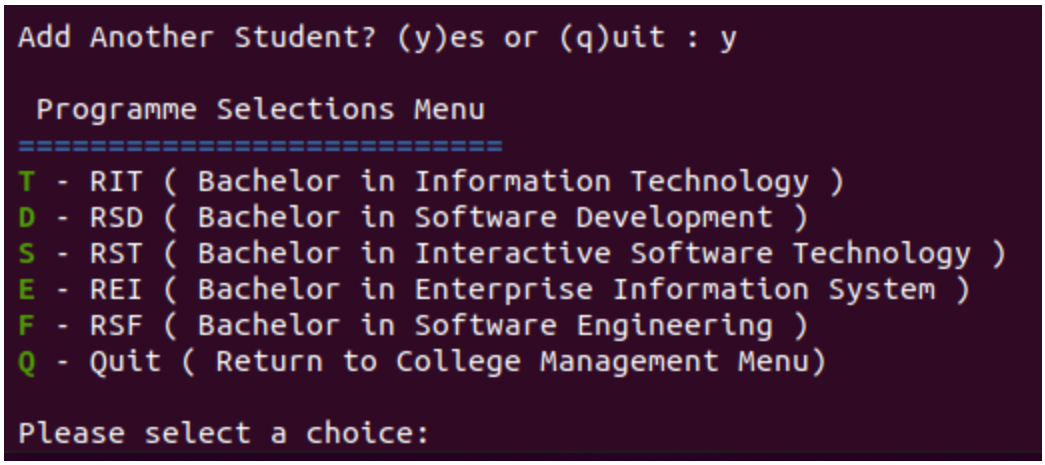


Figure 2.7 Add Another Student

If the user wishes to add another student, the Programme Selections Menu will be displayed.

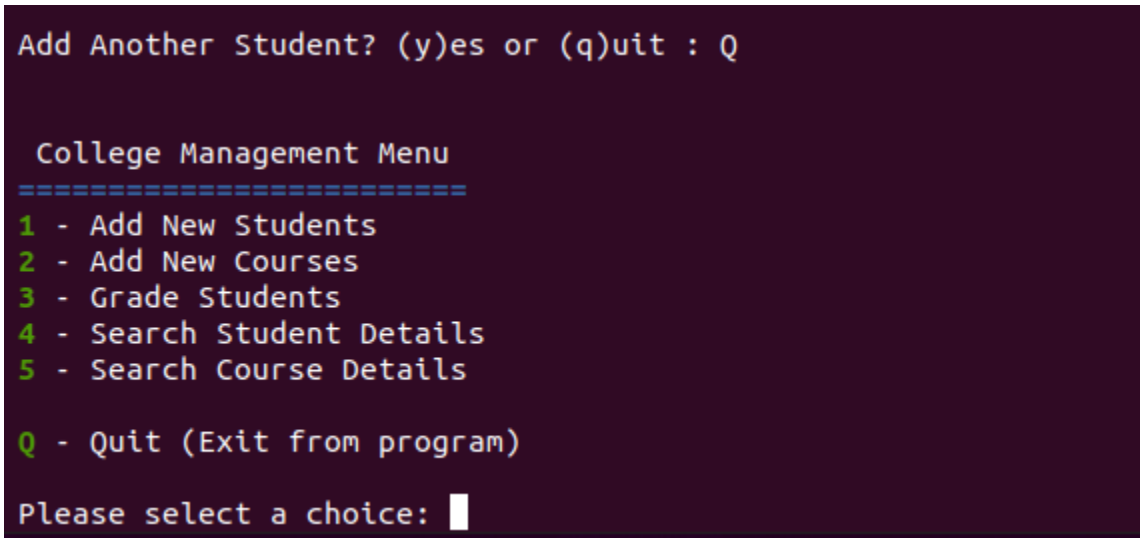


Figure 2.8 Return to College Management Menu

If the user doesn't wish to add another student, the user will input Q and it will return back to the College Management Menu.

Input Validation

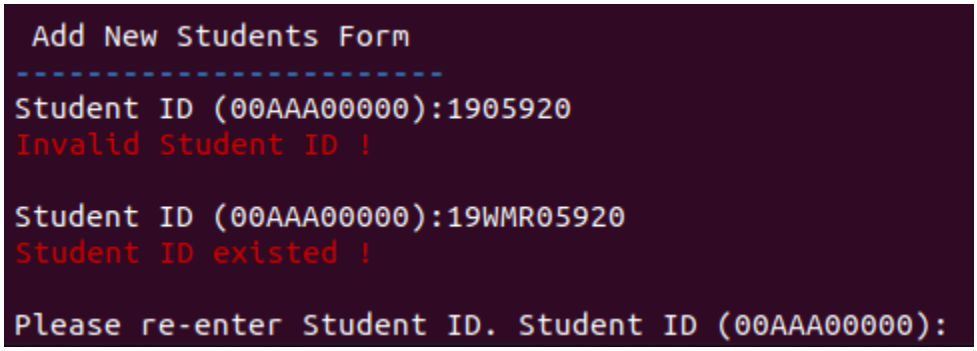


Figure 2.9 Input Validation

If the user does not follow the format of Student ID, the program will prompt an error message "Invalid Student ID". Then, the user is requested to re-enter the Student ID. If the user entered a duplicated Student ID, the program will prompt an error message "Student ID existed". Then, the user is requested to re-enter the Student ID.

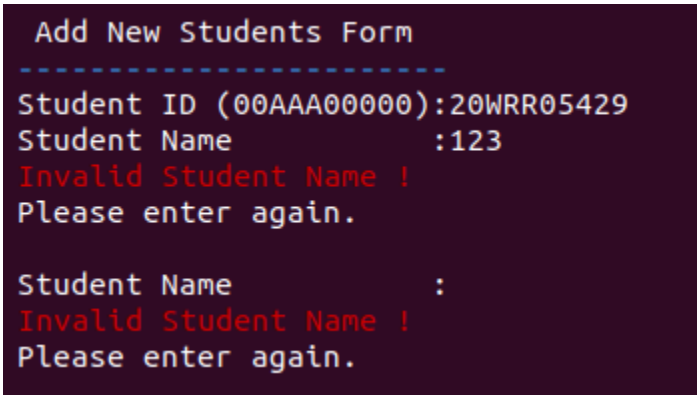


Figure 2.10 Input Validation

If the user enters numbers in Student Name or leave it blank, an error message "Invalid Student Name" will be displayed. Then, user is required to enter the Student Name again.

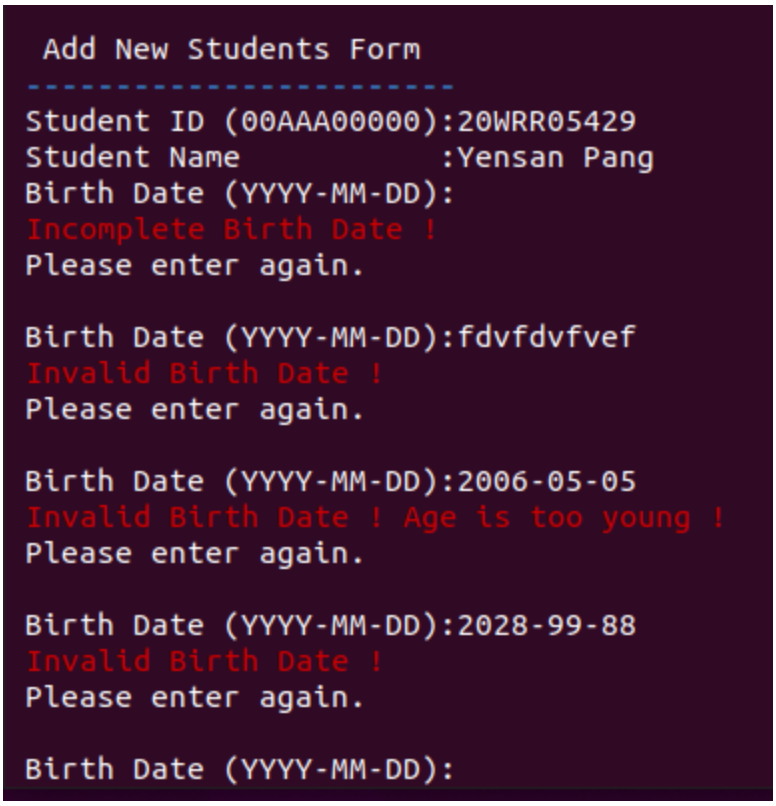


Figure 2.11 Input Validation

User must input a valid date in order to continue the program. Invalid input will cause the system to prompt an error message “Invalid Birth Date” or “Incomplete Birth Date” or “Invalid Birth Date! Age is too young !” will be displayed and the user needs to enter the birth date again. Students must be at least born in 2004 to registered as a student in this College.

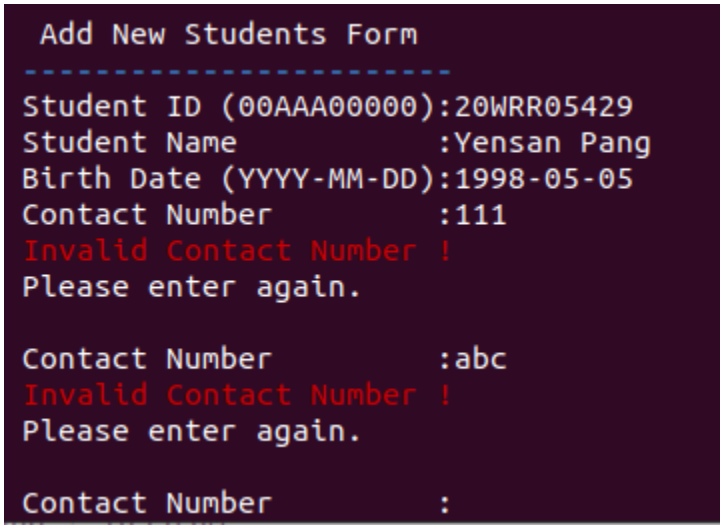


Figure 2.12 Input Validation

The format for Contact Number is 000-0000000000. If the user did not follow, the error message “ Invalid Contact Number” will be shown and the user needs to enter again. The user can’t enter characters in the contact number. Contact Number must be in numerical form.

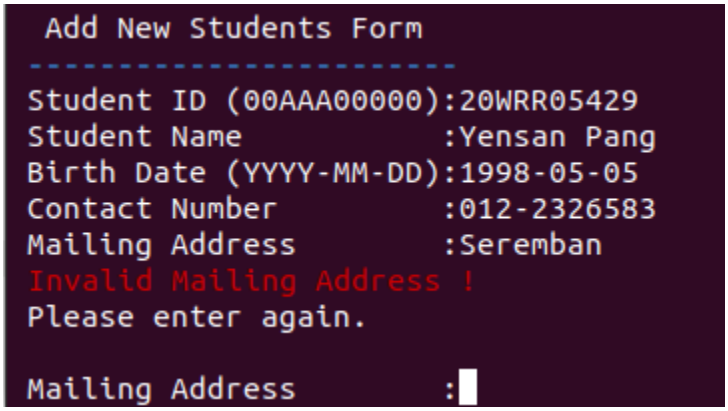


Figure 2.13 Input Validation

The mailing address should be more than 10 characters. If the user input the mailing address less than 10 characters, it will be a invalid input. User required to enter again.

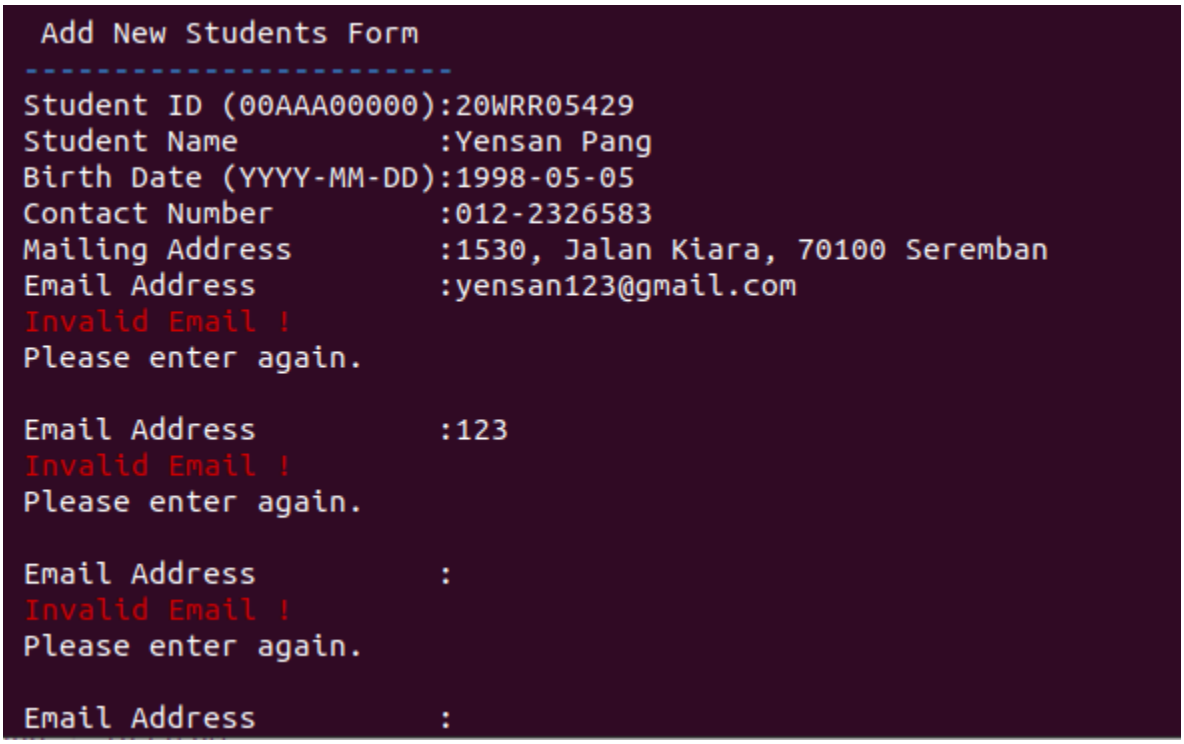


Figure 2.14 Input Validation

The email address should be in the format xxxx@tarc.edu.my. If user input "...@gmail.com" , this will be an invalid email address. The user needs to enter again. "123" is not a valid email address too. User needs to enter again.

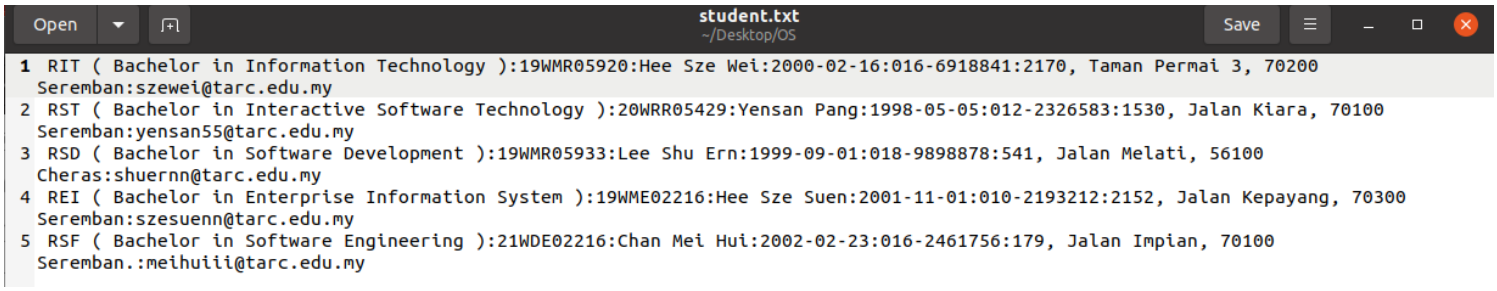


Figure 2.15 student.txt
This is the student.txt where it stores all the student details.

PART B: When the user decides to search for a student details in the system

```
 College Management Menu
=========================
1 - Add New Students
2 - Add New Courses
3 - Grade Students
4 - Search Student Details
5 - Search Course Details

Q - Quit (Exit from program)

Please select a choice: 4
Displaying Search Student Details Form...

 Search Student Details Form
-----------------------------

Enter Student ID [00AAA00000]:
```

Figure 2.18 Search Student Details

The user selected '4' in the College Management Menu to search for student details. Search student details form will then be displayed. User needs to enter an existing Student ID.

```
 Search Student Details Form
-----------------------------

Enter Student ID [00AAA00000]: 19WMR05920
------------------------------------------------------------------------

Student Name    :  Hee Sze Wei
Birth Date      :  2000-02-16
Contact Number  :  016-6918841
Mailing Address :  2170, Taman Permai 3, 70200 Seremban
Email           :  szewei@tarc.edu.my


Search Another Student? (y)es or (q)uit :
Press (q) to return to College Management Menu
```

Figure 2.19 Search Student Details Form

If the user enters an existing student ID, student details will be displayed which retrieve from the student.txt.

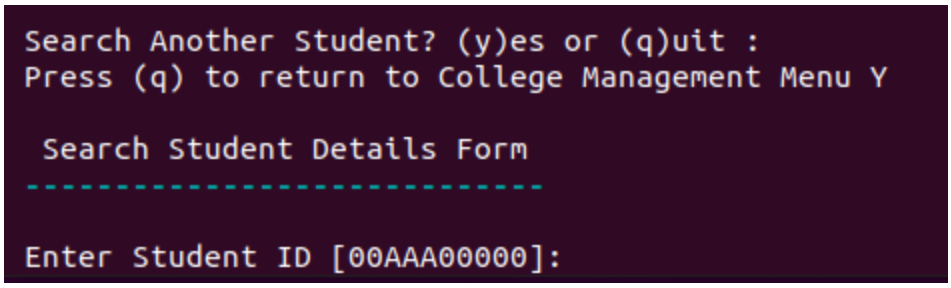


Figure 2.20 Search Another Student

If the user wishes to add another student, the user is required to select the choice 'Y'. After selecting choice 'Y', search student details form will be displayed.

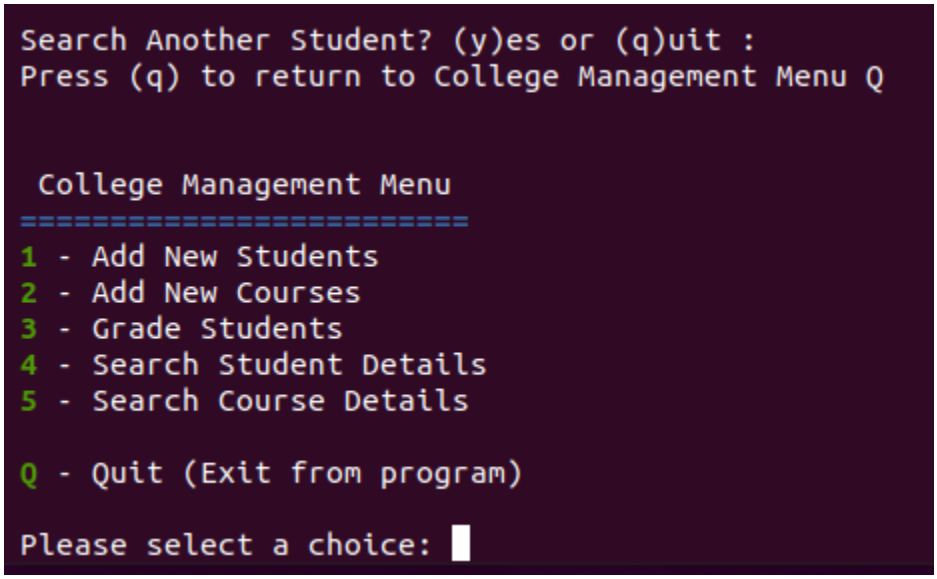


Figure 2.21 Quit Search Student Form

If the user wishes to quit from the search student details form, the user is required to select the choice 'Q'. After selecting 'Q', it will return to the College Management Menu.

Input Validation

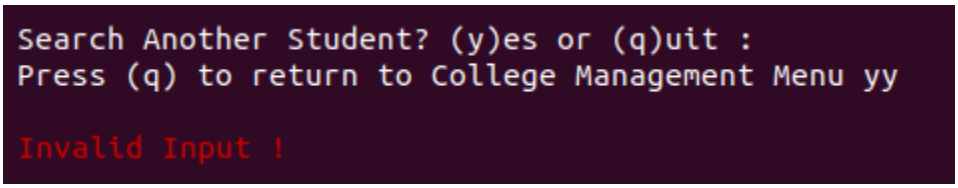


Figure 2.22

turn in liao why u still here

```
Search Another Student? (y)es or (q)uit :
Press (q) to return to College Management Menu 1

Invalid Input !
```

Figure 2.23

Figure 2.22 - 2.23 show the input validation. Users can only enter one y or one q to continue the program. If the user selects choices other than y and q, it will show an error message "Input Validation. Then, the user is requested to select again.

```
 Search Student Details Form
-----------------------------

Enter Student ID [00AAA00000]: 19WEE09098
--------------------------------------------------------------------------------

Student ID:  19WEE09098  does not exist !

Search Another Student? (y)es or (q)uit :
Press (q) to return to College Management Menu
```

Figure 2.24 Input Validation

If the user search a student ID which does not exist, it will tell the user that the student ID does not exist. Then, user can choose to search student again with an existing ID or quit this form and return back to the College Management Menu.

```
 Search Student Details Form
-----------------------------

Enter Student ID [00AAA00000]: weefgbtr
--------------------------------------------------------------------------------

Invalid Student ID !
Please enter again.

Enter Student ID [00AAA00000]:
```

Figure 2.25 Input Validation

Users are required to input a Student ID which follows the right format.

Sample Code
Programme Selections Menu (prog)

```
#!/bin/bash

readonly RIT=" RIT ( Bachelor in Information Technology )"
readonly RSD=" RSD ( Bachelor in Software Development )"
readonly RST=" RST ( Bachelor in Interactive Software Technology )"
readonly REI=" REI ( Bachelor in Enterprise Information System )"
readonly RSF=" RSF ( Bachelor in Software Engineering )"

echo "  Programme Selections Menu"
echo -e "\e[1;34m ========================== \e[0m"

echo -e "\e[1;32m T \e[0m- RIT ( Bachelor in Information Technology ) "
echo -e "\e[1;32m D \e[0m- RSD ( Bachelor in Software Development ) "
echo -e "\e[1;32m S \e[0m- RST ( Bachelor in Interactive Software Technology ) "
echo -e "\e[1;32m E \e[0m- REI ( Bachelor in Enterprise Information System ) "
echo -e "\e[1;32m F \e[0m- RSF ( Bachelor in Software Engineering ) "

echo -e "\e[1;32m Q \e[0m- Quit ( Return to College Management Menu) "
echo " "

validInput=false
while [ $validInput = "false" ]
do
echo -n " Please select a choice: "; read choice
choice=$(echo $choice | tr "[[:lower:]]" "[[:upper:]]" )

 if ! [[ $choice =~ ^[DEFQST]$ ]]
 then
  echo -e "\033[31;5m Invalid Input ! \033[0m"
  echo " Please enter agian."
  validInput=false
 else
   validInput=true
 fi
done

```

```
37 case $choice in
38  [Tt]) prog="$RIT"
39        ;;
40  [Dd]) prog="$RSD"
41        ;;
42  [Ss]) prog="$RST"
43        ;;
44  [Ee]) prog="$REI"
45        ;;
46  [Ff]) prog="$RSF"
47        ;;
48  [Qq]) echo " Returning to College Management Menu..."
49        sleep 1
50        ./ManagementMenu
51        ;;
52 esac
53
54 echo " Programme Selected: $prog "
55 echo -e "\033[33;5m User is requested to fill in the Student Details Form. \033[0m"
56
57 ./addStud "$prog"  #1st argument
58
```

Add New Student Form(addStud)

```
#!/bin/bash
prog=$1

studList="/home/sereneee/Desktop/OS/student.txt"
echo " "
echo "  Add New Students Form"
echo -e "\e[1;34m ----------------------- \e[0m"

validID=false
while [ $validID = "false" ]

do
 echo -n " Student ID (00AAA00000):"; read studID
 studID=$(echo $studID | tr "[[:lower:]]" "[[:upper:]]" )

 if [ -z "$studID" ] || ! [[ $studID =~ ^[0-9][0-9][A-Z][A-Z][A-Z][0-9][0-9][0-9][0-9][0-9]$ ]]
 then
  echo -e "\033[31;5m Invalid Student ID ! \033[0m"
  echo " "
  validID=false
 else
  if grep -q $studID $studList ;
  then
    studentRecord=$(grep $studID $studList)
    IFS=: read programmes IDS names dates hp addresses email <<< $studentRecord
        echo -e "\033[31;5m Student ID existed ! \033[0m"
        echo " "
        echo -n " Please re-enter Student ID."
        validID=false
   else
        validID=true
   fi
 fi
done

#-----------------------------------------
while [ "$studName" = "" ] || ! [[ $studName =~ [A-Za-z]{2,40} ]] ;
do
 echo -n " Student Name                :" ; read studName
 if [ -z "studName" ] || ! [[ $studName =~ [A-Za-z]{2,40} ]]
 then
   echo -e "\033[31;5m Invalid Student Name ! \033[0m"
   echo " Please enter again."
   echo " "
 fi
done

#-----------------------------------------
minAge=$(date -d 2004-01-01 +"%Y%m%d")
validDate=false
while [ $validDate = "false" ]
do
 echo -n " Birth Date (YYYY-MM-DD):"; read date
 if ! [ -z $date ]
 then
   if [ $(date -d "$date" +"%Y%m%d" 2>> /dev/null) ]
    then
     if [[ "$date" < "$minAge" ]]
      then
       validDate=true
    else
       echo -e "\033[31;5m Invalid Birth Date ! Age is too young ! \033[0m"
       echo " Please enter again."
       echo " "
       validDate=false
    fi
   else
    echo -e "\033[31;5m Invalid Birth Date ! \033[0m"
       echo " Please enter again."
       echo " "
       validDate=false
   fi
 else
   echo -e "\033[31;5m Incomplete Birth Date ! \033[0m"
   echo " Please enter again."
   echo " "
   validDate=false
```

```
 fi
done


#----------------------------------------
while [ "$contactNum" = "" ] || ! [[ $contactNum =~ ^[0][1][0-9]-[0-9]{7,8}$ ]] ;
do
 echo -n " Contact Number             :"; read contactNum
 if [ -z "$contactNum" ] || ! [[ $contactNum =~ ^[0][1][0-9]-[0-9]{7,8}$ ]]
 then
   echo -e "\033[31;5m Invalid Contact Number ! \033[0m"
   echo " Please enter again."
   echo " "
 fi
done

#----------------------------------------
while [ -z "$mailAdd" ] || ! [[ $mailAdd =~ ^[[:graph:][:space:]]{10,40}$ ]]
do
 echo -n " Mailing Address            :" ; read mailAdd
 if [ -z "$mailAdd" ] || ! [[ $mailAdd =~ ^[[:graph:][:space:]]{10,40}$ ]]
 then
   echo -e "\033[31;5m Invalid Mailing Address ! \033[0m"
   echo " Please enter again."
   echo " "
   #validMail=false
 #else
   #validMail=true
 fi
done

#----------------------------------------
while [ -z "$email" ] || ! [[ "$email" =~ ^[[:alnum:]]{6,20}\@tarc.edu.my$ ]]
do
 echo -n " Email Address              :"; read email
 if [ -z "$email" ] || ! [[ "$email" =~ ^[[:alnum:]]{6,20}\@tarc.edu.my$ ]]
 then
   echo -e "\033[31;5m Invalid Email ! \033[0m"
   echo " Please enter again."
   echo " "
 fi
done

#----------------------------------------
echo " Saving..."
sleep 2
echo "$prog:$studID:$studName:$date:$contactNum:$mailAdd:$email" >> student.txt

#----------------------------------------
choice=n
while [ "$choice" != "y" ] || [ "$choice" != "q" ]
do
 echo " "
 echo -n  " Add Another Student? (y)es or (q)uit : "; read choice
 echo " "
 case "$choice" in
 [Yy]) ./prog;
       exit;;
 [Qq]) ./ManagementMenu;
       exit;;
 *) echo -e "\033[31;5m Invalid Input ! \033[0m"
    echo " Please enter again." ;;
 esac
done
```

Search Student Details Form(searchStud)

```
#!/bin/bash

stud_file="/home/sereneee/Desktop/OS/student.txt"

echo "  Search Student Details Form"
echo -e "\e[1;36m ---------------------------- \e[0m"
echo " "

 validCode=f
 while [ $validCode = "f" ]
  do
   echo -n " Enter Student ID [00AAA00000]: "; read studID
   echo -e "\e[1;34m ____________________________________________________________________ \e[0m"
   echo " "

   studID=${studID^^} #Change to uppercase


   if [ -n $studID ] && [[ $studID =~ ^[0-9][0-9][A-Z][A-Z][A-Z][0-9][0-9][0-9][0-9][0-9]$ ]]
        then
        validCode=t
   else
        echo -e "\033[31;5m Invalid Student ID ! \033[0m"
        echo " Please enter again."
        echo " "
        validCode=f
   fi
  done

  result="$(grep -i "$studID" "$stud_file")"

  while IFS=: read -r prog id name date contact mailAdd email
   do
        if [[ -n $prog && -n $id && -n $name && -n $date && $contact && -n $mailAdd && -n $email ]]
        then

        printf '%-17s%-3s%-17s\n' " Student Name " ":" "$name"
        printf '%-17s%-3s%-17s\n' " Birth Date " ":" "$date"
        printf '%-17s%-3s%-17s\n' " Contact Number " ":" "$contact"
        printf '%-17s%-3s%-17s\n' " Mailing Address " ":" "$mailAdd"
        printf '%-17s%-3s%-17s\n' " Email " ":" "$email"
        echo " "

        else
        #echo " This Student ID $studID does not exist !"
        echo -e " Student ID: \e[1;33m $studID \e[0m does not exist !"
        fi
  done <<< $result

while [ "$choice" != "y" ] || [ "$choice" != "q" ]
do
 echo " "
 echo " Search Another Student? (y)es or (q)uit : "
 echo -n  " Press (q) to return to College Management Menu " ; read choice
 echo " "
 case "$choice" in
 [Yy]) ./searchStud;
        exit;;
 [Qq]) ./ManagementMenu;
        exit;;
 *) echo -e "\033[31;5m Invalid Input ! \033[0m" ;;
 esac
done
```

Task 3:

Screen Output (s)

PART A: When user select to add new course

```
 College Management Menu
==========================
1 - Add New Students
2 - Add New Courses
3 - Grade Students
4 - Search Student Details
5 - Search Course Details

Q - Quit (Exit from program)

Please select a choice: 2
Displaying Add New Courses Form...
```

Figure 3.1

Users select "2" to proceed for adding a new course into the system.

```
Add new Course Form
===================

Course Code [BAxx0000]: BACS1234
Course Name: Information Technology
Credit Hours(1-12): 4

Saving ......

Add Another Course? (y)es or (q)uit:
Press (q) to return to College Management Menu
```

Figure 3.2 Add New Course Form

This part is adding a new course into the system by inputting a valid course code, course name and the credit hours.

Input Validation

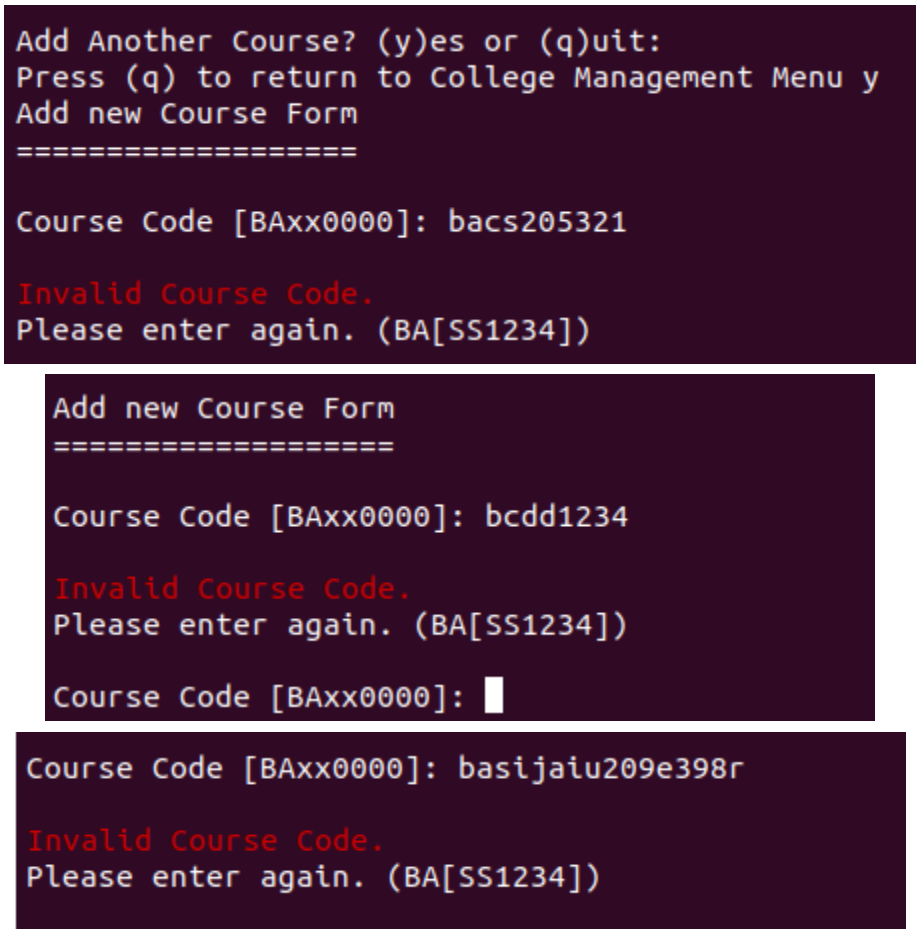


Figure 3.3 - 3.5 shows the input validation for the course code.
The system will pop out the error message when the user input the invalid format of the course code.

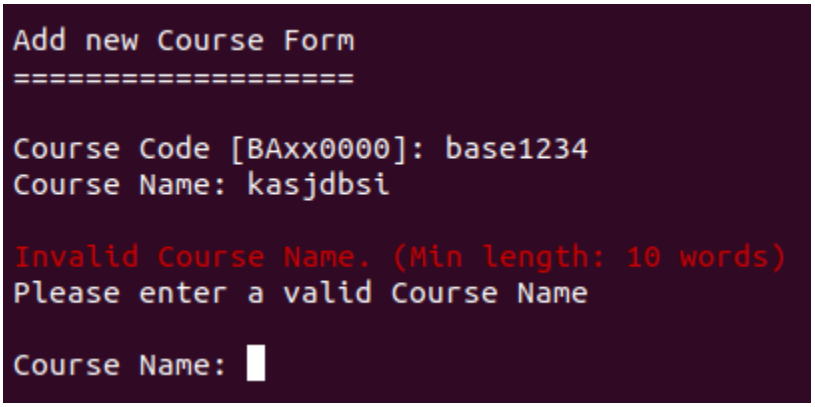
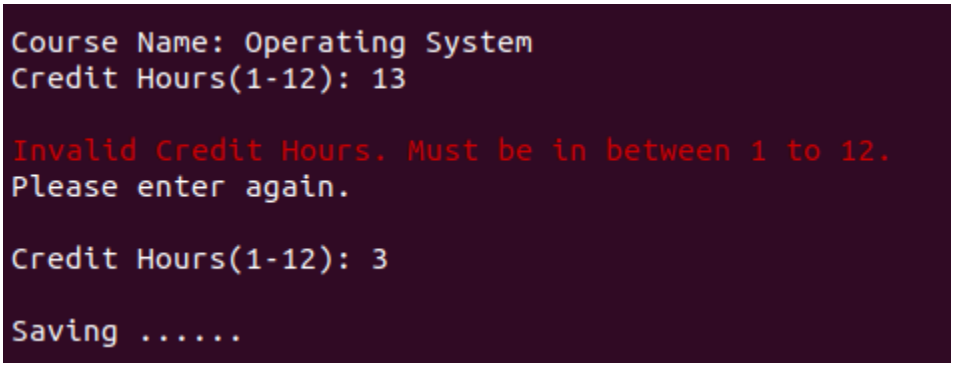
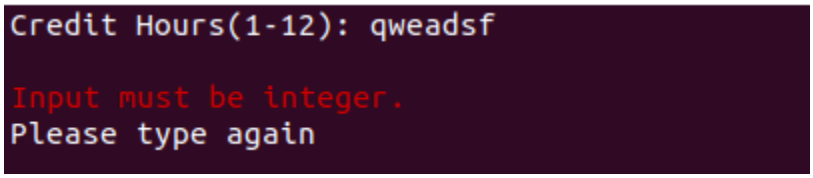


Figure 3.6 Input Validation
Error messages will pop out when the user input the course name with the length of less than

10 characters.





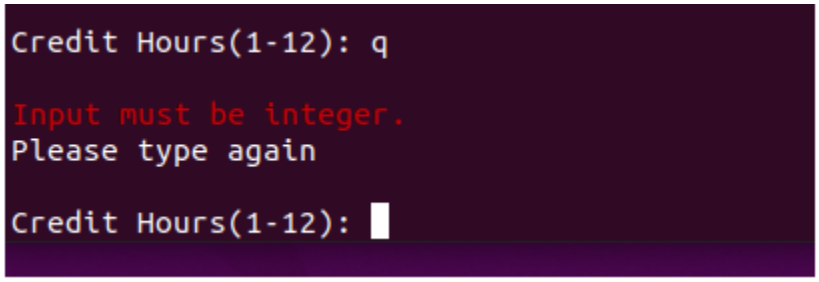
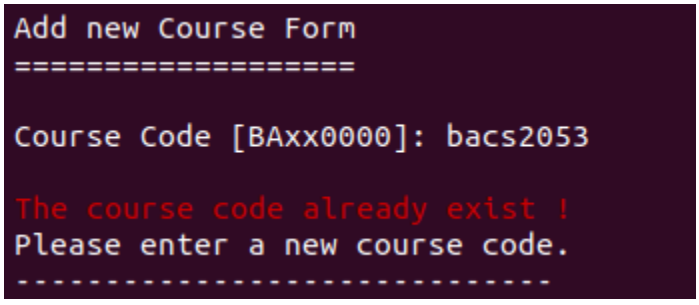


Figure 3.7 - 3.9 shows the input validation for the credit hours.
The system will pop out an error message when the user input the value of credit hours which is out of the range, input a non numerical value or input a sentence.

Check if there any duplicate data

- To avoid duplicate data being added into the system file



```
Please enter a new course code.
------------------------------
Course Code [BAxx0000]: bacs2093
Course Name: information TecHnology

Duplicate Course Name !
Please enter a new Course Name.

Course Name: Operating System
```

Figure 3.10 - 3.11 shows the error message will pop out once the system found the same course code and course name in the system file.

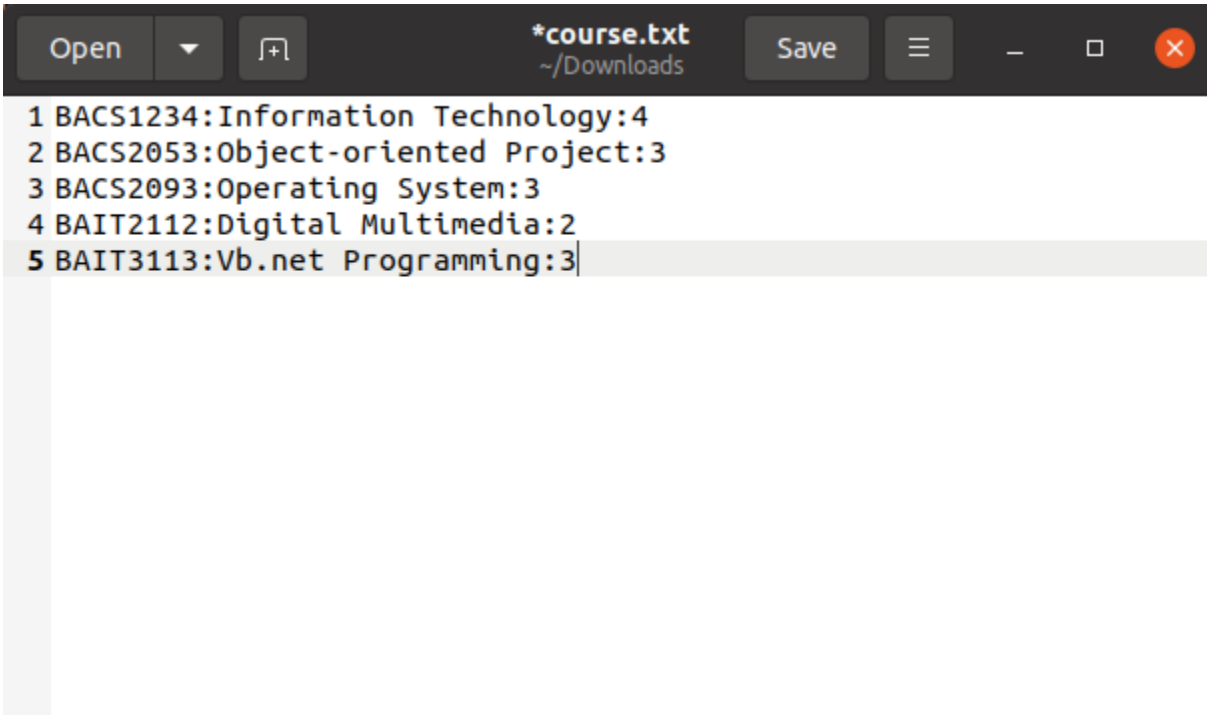


Figure 3.12 course.txt

It is a text file named “course.txt” that stores all the current course and new course details.

PART B: When the user decides to search for a course in the system

```
Please select a choice: 5
Displaying Search Course Details Form...

 Search Course Details Form
----------------------------

Enter Course Code: BACS1234
------------------------------------------------------------------------------

Course Name     :  Information Technology
Credit Hours    :  4


Search Another Course? (y)es or (q)uit:
Press(q) to return to College Management Menu. y
```

Figure 3.13 Search Course Details Form

Users enter this search course details form after selecting "5" from the management menu section. Users search for the course details by entering the course code. Once the system successfully finds out the course details, it will display them on the output.

```
Search Another Course? (y)es or (q)uit:
Press(q) to return to College Management Menu. y

 Search Course Details Form
----------------------------

Enter Course Code: bait2112
------------------------------------------------------------------------------

Course Name     :  Digital Multimedia
Credit Hours    :  2
```

Figure 3.14 shows that the system is able to proceed to the search course page after user input 'y'.

After the user successfully found the course details, the system will prompt a message to ask whether the user wants to continue searching for another different course code or back to the main menu.

Input Validation

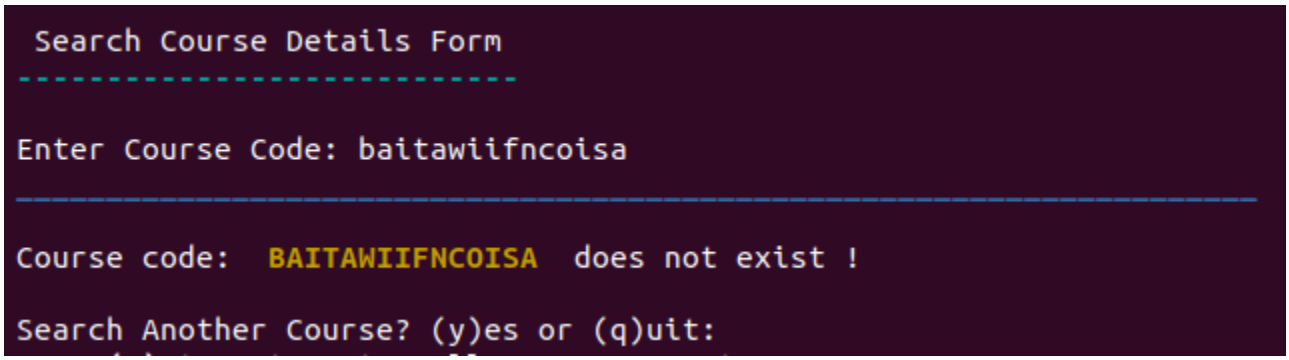


Figure 3.15 Input Validation

If a user enters a wrong format of course code or the course code that does not exist, it will be considered as a course code that does not exist in the course.txt file . Then, an error message will be displayed.

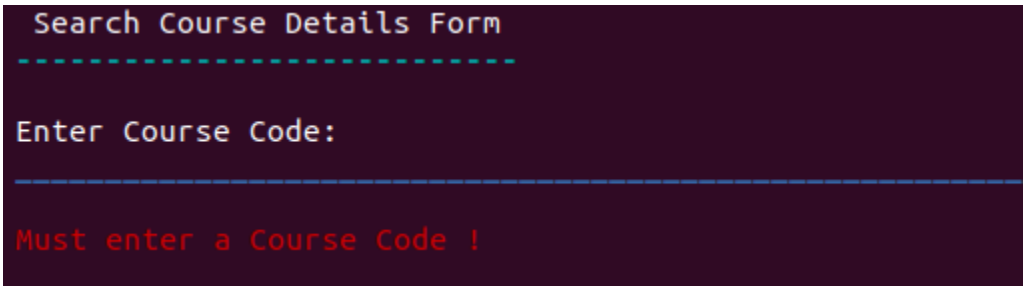


Figure 3.16 Input Validation

The input must not be empty.

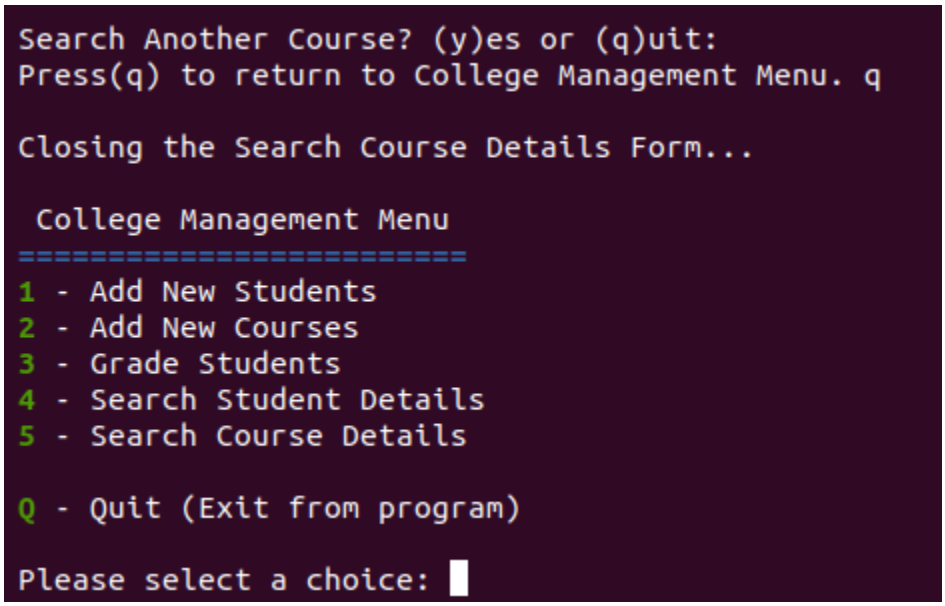


Figure 3.17 Quit to Main Menu (Management Menu)

Sample Code
Add New Course Form(addCourse)

```
#!/bin/bash

course_file="/home/sereneee/Desktop/OS/course.txt"

echo " Add new Course Form"
echo " ==================="
echo " "

# Search avoid duplicate course number
course_file="/home/sereneee/Desktop/OS/course.txt"

validCode=f #false
while [ $validCode = "f" ]
 do
  echo -n " Course Code [BAxx0000]: " ; read courseCode
  courseCode=${courseCode^^} #Change to uppercase

# The input need to be in format such as begin with BA

  if [ -n $courseCode ] && [[ $courseCode =~ ^BA[A-Z][A-Z][0-9][0-9][0-9][0-9]$ ]]
  then
    # Check for duplicate # q - no output
    if grep -q  $courseCode $course_file
    then
      echo " "
      echo -e "\033[31;5m The course code already exist ! \033[0m"
      echo " Please enter a new course code."
      echo " ------------------------------"
      validCode=f
    else
      validCode=t
    fi
  else
    echo " "
    echo -e "\033[31;5m Invalid Course Code. \033[0m"
    echo " Please enter again. (BA[SS1234])"
    echo " "
    validCode=f
  fi
done
```

```
41
42 # Check for null string
43 validName=f
44  while [ $validName = "f" ]
45  do
46   echo -n " Course Name: " ; read courseName
47   name=$(echo $courseName | tr -d '[[:space:]]' )
48
49   if [ -z $name ]
50   then
51     echo " "
52     echo -e "\033[31;5m Empty input ! \033[0m"
53     echo " Please enter valid input."
54     echo " "
55     validName=f
56   else
57     validName=t
58     courseName=$(echo $courseName | tr -s " " | tr [[:upper:]] [[:lower:]] ) #Remove extra space
59
60     courseName=($courseName)
61     courseName="${courseName[@]^}"
62
63 # Variable stored tha input course name and the space between had been removed
64 # name=$(echo $courseName | tr -d '[[:space:]]' )
65
66    if [[ $name =~ [A-Za-z\-]{10,} ]]
67     then
68      # Check duplicate name # Igonre the space in course.txt and also input
69      if cat $course_file | tr -d '[[:space:]]' | grep -i -q -w $name
70       then
71         echo " "
72         echo -e "\033[31;5m Duplicate Course Name ! \033[0m"
73         echo " Please enter a new Course Name."
74         echo " "
75         validName=f
76      fi
77    else
78      echo " "
79      echo -e "\033[31;5m Invalid Course Name. (Min length: 10 words) \033[0m"
80      echo " Please enter a valid Course Name"
81      echo " "
82      validName=f
83    fi
84   fi
85  done
86
87  validHours=f
88 # Check for valid input number
89  while [ $validHours = "f" ] # if credit hours = string
90  do
91   echo -n " Credit Hours(1-12): " ; read creditHours
92    if [[ $creditHours =~ ^[0-9]+$ ]] # Check is integer or not
93     then # integer
94      if [ $creditHours -ge 1 -a $creditHours -le 12 ]
95        then
96          validHours=t
97      else # for the credit hours that are not in range
98        echo " "
99         echo -e "\033[31;5m Invalid Credit Hours. Must be in between 1 to 12. \033[0m"
100       echo " Please enter again."
101       echo " "
102       validHours=f
103     fi
```

```
  else # if the credit hours is string
    echo " "
    echo -e "\033[31;5m Input must be integer. \033[0m"
    echo " Please type again"
    echo " "
    validHours=f
  fi
done

echo " "
echo " Saving ......"
sleep 3
echo "$courseCode:$courseName:$creditHours" >> course.txt
echo " "

# To check whether input a valid letter (y and q)
respond=n
while [[ $respond =~ ^[^Y|y|Q|q]$ ]]
do
echo " Add Another Course? (y)es or (q)uit: "
echo -n " Press (q) to return to College Management Menu "  ; read respond

if [[ $respond =~ ^[^Y|y|Q|q]$ ]]
 then
   echo " "
   echo -e "\033[31;5m Invalid Input. \033[0m"
   echo " Please enter again [y]yes/[q]quit"
   echo " "
fi
done

case $respond in
[Qq])./ManagementMenu
     ;;
[Yy])./addCourse
     ;;
esac
```

Search Course Detail Form(searchCourse)

```
#!/bin/bash
course_file="/home/sereneee/Desktop/OS/course.txt"

echo "  Search Course Details Form"
echo -e "\e[1;36m ---------------------------- \e[0m"
echo " "

nullInput=t
while [ $nullInput = "t" ]
do
  echo -n " Enter Course Code: " ; read courseCode
  echo -e "\e[1;34m ____________________________________________________________________________ \e[0m"
  echo " "

  if [ -z $courseCode ]
  then
     echo -e "\033[31;5m Must enter a Course Code ! \033[0m"
     echo " "
     nullInput=t
  else
     nullInput=f
  fi
done

courseCode=$(echo $courseCode | tr -s "" | tr "[[:lower:]]" "[[:upper:]]" )

# Save the grep result in the variable
  result="$(echo $courseCode | grep -w "$courseCode" "$course_file")"

  while IFS=: read -r code name hours
   do
    if [[ -n $code && -n $name && -n $hours ]]
     then

       printf '%-17s%-3s%-17s\n' " Course Name " ":" "$name"
       printf '%-17s%-3s%-17s\n' " Credit Hours " ":" "$hours"

       echo " "
    else

      echo -e " Course code: \e[1;33m $courseCode \e[0m does not exist !"

    fi
  done <<< $result
```

```
45
46 #-----------------------------------------------------
47 echo " "
48
49 validRespond=f
50 while [ $validRespond = "f" ]
51 do
52  echo " Search Another Course? (y)es or (q)uit: "
53  echo -n " Press(q) to return to College Management Menu. " ; read respond
54
55  if [[ $respond =~ ^[Y|y|Q|q]$ ]]
56  then validRespond=t
57  else
58    echo -e "\033[31;5m Invalid Input ! \033[0m"
59    echo " Please enter (y)es or (q)quit."
60    validRespond=f
61  fi
62 done
63
64 echo " "
65
66 case $respond in
67 [Yy]) ./searchCourse
68       exit
69       ;;
70 [Qq]) echo " Closing the Search Course Details Form..."
71       sleep 1
72       ./ManagementMenu
73       ;;
74 esac
```

Task 4:

Screen Output(s)

```
Please select a choice: 3
Displaying Student Validation Form...

 Student Validation Form
--------------------------
Enter Student ID [00XXX00000]: 19WMR05920

-----------------------------------------------------------------

Student Name  :  Hee Sze Wei
Programme     :  RIT ( Bachelor in Information Technology )

-----------------------------------------------------------------

Press (n) to continue to enter the student's marks:
(q) to return to College Management Menu:
```

Figure 4.1 Student Validation Form

Student validation form will be displayed when the user selects '3' at the main menu section. This form is used to validate the student name and the programme that student takes by using the student id number. Once a valid correct student id is entered, the student name and programme will be displayed.

```
Press (n) to continue to enter the student's marks:
(q) to return to College Management Menu:n

 Student Examination Marks Form
--------------------------------
Academic Year (YYYY):
```

Figure 4.2

Users are able to proceed to insert the examination marks for the student when users select 'n' at the validation form.

```
 Student Examination Marks Form
--------------------------------
Academic Year (YYYY): 2020
Semester (1 / 2 / 3 ): 1
--------------------------------------------------------------------------------


File Name: 19WMR05920Result.txt

Please enter the course code:
```

Figure 4.3 Student Examination Marks Form

Users are able to directly proceed to this student examination marks form to insert the examination marks for a particular academic year and semester of a student.

```
--------------------------------------------------------------------------------


File Name: 19WMR05920Result.txt

Please enter the course code: BACS1234
Enter the marks obtained: 70

Saving the student examination marks...

-----------------------------------
```

Figure 4.4 shows that the student examination result will be saved in a text file which is named with the student ID.

Users are required to know about which course the student takes and then input the marks for that particular course.

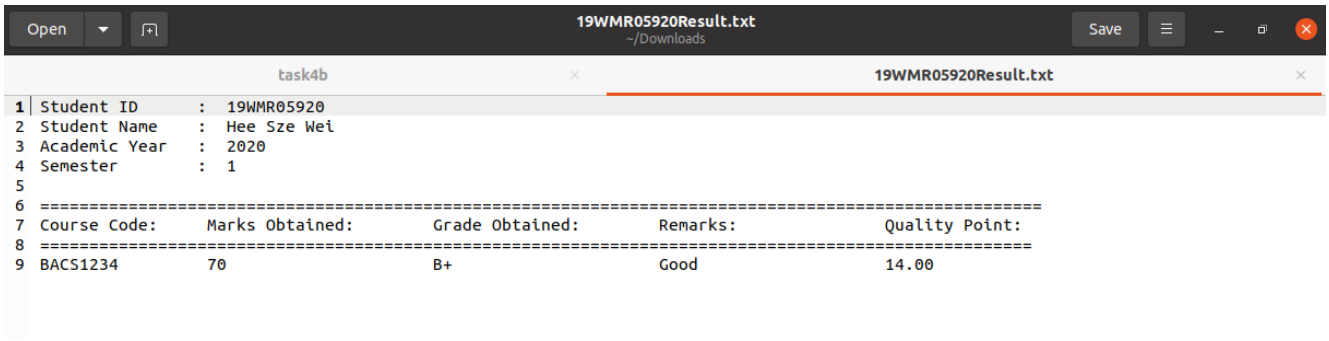


Figure 4.5 19WMR05920Result.txt file

This text file shows the details of the student with their examination results in a text file.

```
-----------------------------------

Press (y) to continue to enter the student's marks or (b) to back previous or (m) back to Management Menu:
```

Figure 4.6

After inputting the examination marks for a subject, user prompts the message that asked whether the user wanted to continue inserting the result for the particular student or back to the student validation form to enter the examination marks for another student or required to quit this form.

```
Press (y) to continue to enter the student's marks or (b) to back previous or (q) back to Management Menu: b


Back to Student Validation Form...

 Student Validation Form
--------------------------
Enter Student ID [00XXX00000]:
```

Figure 4.7

If users choose to go back to the Student validation form [b] or back to the main menu [q], the system will automatically compute the result for the student that the user had just entered. The result will be shown in the Student Result Text file.

Open | 19WMR05920Result.txt ~/Downloads | Save

task4b | 19WMR05920Result.txt

```
1  Student ID      :  19WMR05920
2  Student Name    :  Hee Sze Wei
3  Academic Year   :  2020
4  Semester        :  1
5
6  ============================================================================================
7  Course Code:     Marks Obtained:        Grade Obtained:        Remarks:          Quality Point:
8  ============================================================================================
9  BACS1234         70                     B+                     Good              14.00
10                                                                 Total Quality Point:   14.00
11
```

Figure 4.8 Example of Student result text file

Open | 20WRR05429Result.txt ~/Downloads | Save

course.txt | 20WRR05429Result.txt

```
1  Student ID      :  20WRR05429
2  Student Name    :  Yensan Pang
3  Academic Year   :  2021
4  Semester        :  2
5
6  ============================================================================================
7  Course Code:     Marks Obtained:        Grade Obtained:        Remarks:          Quality Point:
8  ============================================================================================
9  BACS1234         70                     B+                     Good              14.00
10 BAIT2112         90                     A                      Execellent        8.00
11 BAIT3113         40                     F                      Failed            0.00
12                                                                 Total Quality Point:   22.00
13
```

Figure 4.9 Example of Student Result text file

Student Result text file named based on the student id. This text file shows the details of the examination result and total quality point for the marks that were entered by the user at the previous page.

Open 19WMR05933Result.txt ~/Downloads Save

```
Student ID      :  19WMR05933
Student Name    :  Lee Shu Ern
Academic Year   :  2021
Semester        :  1

=============================================================================================================
Course Code:     Marks Obtained:        Grade Obtained:        Remarks:               Quality Point:
=============================================================================================================
BACS1234         90                     A                      Execellent             16.00
BACS2093         70                     B+                     Good                   10.50
                                                               Total Quality Point:   26.50

Student ID      :  19WMR05933
Student Name    :  Lee Shu Ern
Academic Year   :  2021
Semester        :  2

=============================================================================================================
Course Code:     Marks Obtained:        Grade Obtained:        Remarks:               Quality Point:
=============================================================================================================
BACS2053         70                     B+                     Good                   10.50
BAIT3113         90                     A                      Execellent             12.00
                                                               Total Quality Point:   22.50
```

Figure 4.10 shows the different output of results of students based on the different sessions of academic year and semester.

Input Validation at Student Validation Form

```
 Student Validation Form
---------------------------
Enter Student ID [00XXX00000]: werstcymvbuhfjnk


Invalid Student ID !
Please enter again.
```

```
Enter Student ID [00XXX00000]: wqvf3qr egbf fe


Invalid Student ID !
Please enter again.

Enter Student ID [00XXX00000]:
```

```
 Student Validation Form
---------------------------
Enter Student ID [00XXX00000]: 1243568


Invalid Student ID !
Please enter again.

Enter Student ID [00XXX00000]:
```

Figure 4.11 - 4.13 Input Validation
If users entered incorrect format of the student id, it is considered as the invalid input and error message will be displayed.

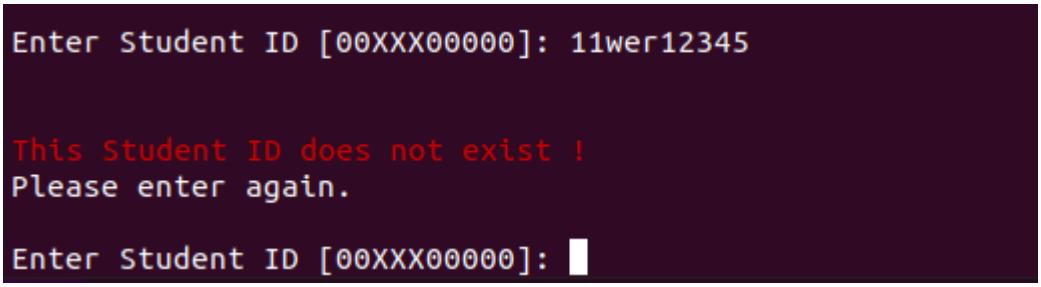


Figure 4.14 Input Validation
If the user entered a correct student id, the system will go through the student.txt to check whether the student id exists or not. Error messages will pop out if the given student id does not exist.

Input Validation for Academic Year

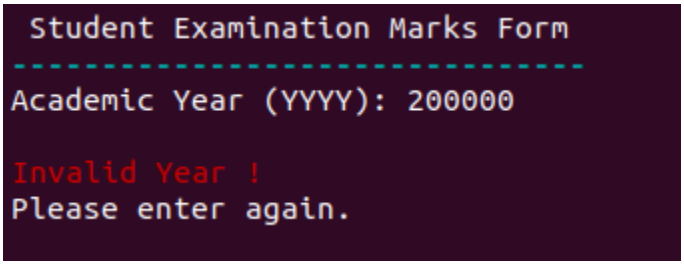


Figure 4.15

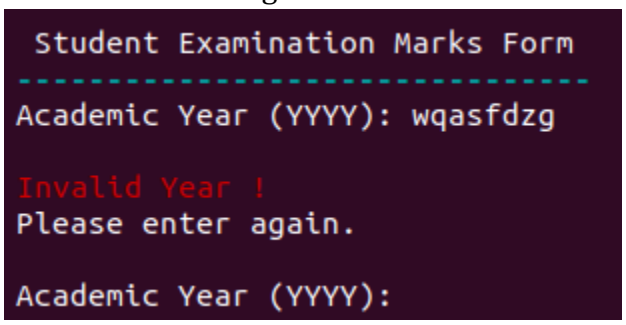


Figure 4.16

Figure 4.15 - 4.16 shows the input validation of the academic year.
Error messages will pop out when the users input the invalid academic year format.

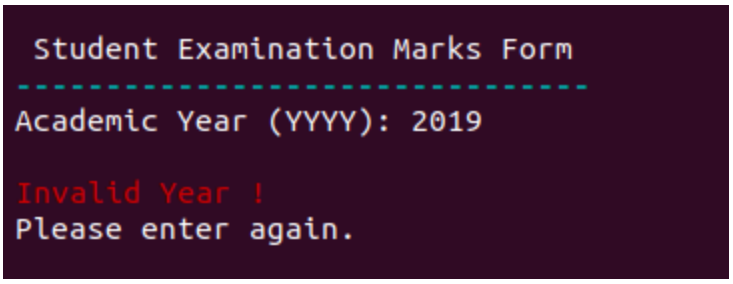


Figure 4.17

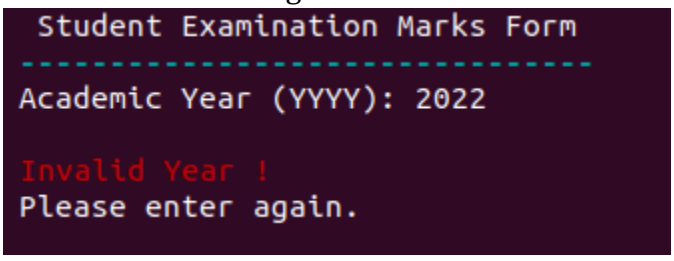


Figure 4.18

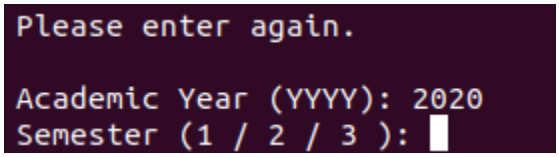


Figure 4.19

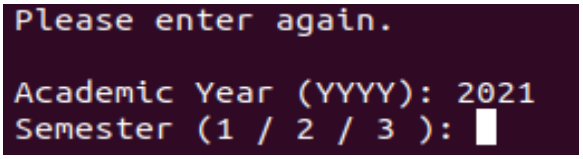
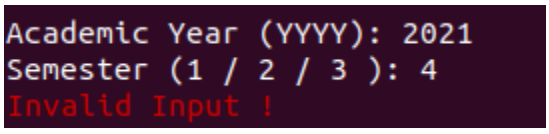


Figure 4.20

Figure 4.17 - 4.20 shows that users are only valid to input the academic year at the range of year 2020 and current year (year 2021).

Input Validation at semester part



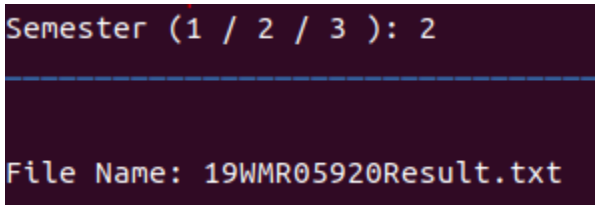


Figure 4.21 - 4.22 shows that users are only allowed to input the semester at the range of 1 to 3.

Course Code validation

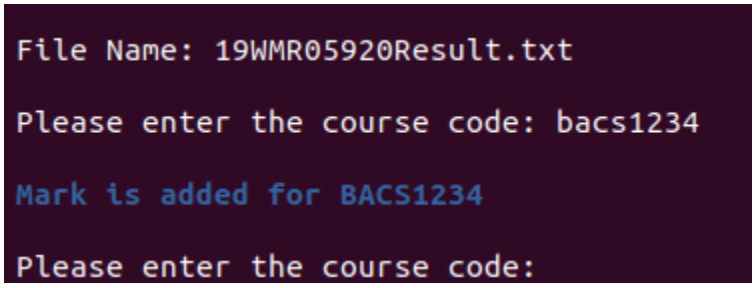


Figure 4.23 shows that error messages will be displayed when the marks for the entered course are already being inserted by the users.

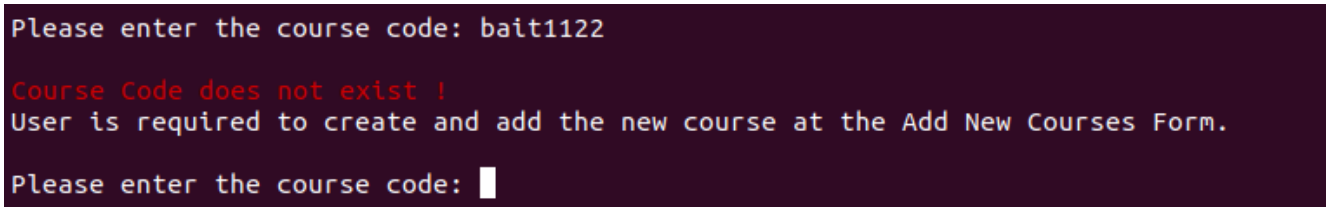


Figure 4.24 shows that error messages pop out if the course code does not exist.

Input Marks Validation

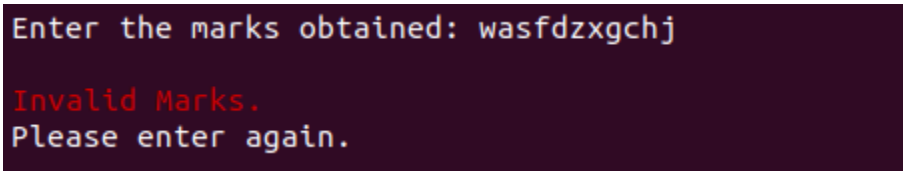


Figure 4.25

Error messages will pop out when users input an invalid format of marks. The examination marks should be in numerical value.

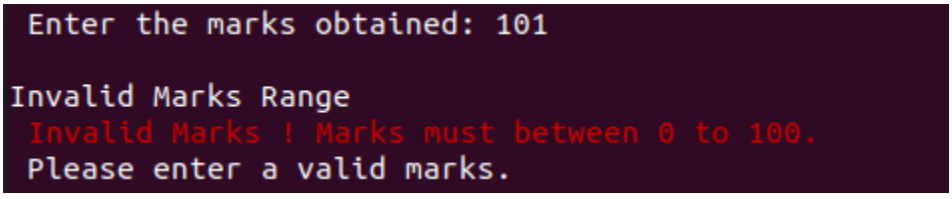


Figure 4.26

The examination marks should be in the range of 1 to 100. Once users enter the marks which are out of the range, error messages will pop out to alert the user.

Duplicate academic year and semester

```
 Student Examination Marks Form
--------------------------------
Academic Year (YYYY): 2021
Semester (1 / 2 / 3 ): 1
--------------------------------------------------------------------------------

The result for year  2021  sem  1  is existed.
```

Figure 4.27
A message will be displayed to inform the user that the result for the input year and semester already existed.

```
You Want to add new course marks for this semester (y)es / (n)o?
```

Figure 4.28 shows that required to input 'y' to continue the next step or input 'n' to exit from this page and back to the Student Validation Form

```
The result for year  2021  sem  1  is existed.

You Want to add new course marks for this semester (y)es / (n)o?y

Please enter the course code for checking purpose:
```

Figure 4.29 shows that users are required to input the course code first before proceeding to add the examination marks. This step is to avoid redundant course examination marks in the same academic year and semester.

```
Please enter the course code for checking purpose: bacs1234

Mark is added for BACS1234
Back to Student Validation...

 Student Validation Form
-------------------------
Enter Student ID [00XXX00000]:
```

Figure 4.30 shows a message pop out when the system found there may be duplicate records if users wanted to insert the examination marks for the input course code. After that will go directly back to the Student Validation Form.

```
You Want to add new course marks for this semester (y)es / (n)o?y

Please enter the course code for checking purpose: BAIT2112

File Name: 19WMR05933Result.txt

Please enter the course code: bait2112
Enter the marks obtained: 80

Saving the student examination marks...

-----------------------------------

Press (y) to continue to enter the student's marks or (b) to back previous or (q) back to Management Menu:
```

Figure 4.31

If the new course code marks are allowed to add into the existing academic year and sem, they will proceed to new step which is adding the record of marks into the result text file,

Open | 19WMR05933Result.txt ~/Downloads | Save

```
Student ID      :  19WMR05933
Student Name    :  Lee Shu Ern
Academic Year   :  2021
Semester        :  1

=============================================================================================================
Course Code:     Marks Obtained:        Grade Obtained:        Remarks:               Quality Point:
=============================================================================================================
BACS1234         90                     A                      Execellent             16.00
BACS2093         70                     B+                     Good                   10.50
                                                               Total Quality Point:   26.50

Student ID      :  19WMR05933
Student Name    :  Lee Shu Ern
Academic Year   :  2021
Semester        :  2

=============================================================================================================
Course Code:     Marks Obtained:        Grade Obtained:        Remarks:               Quality Point:
=============================================================================================================
BACS2053         70                     B+                     Good                   10.50
BAIT3113         90                     A                      Execellent             12.00
                                                               Total Quality Point:   22.50

Student ID      :  19WMR05933
Student Name    :  Lee Shu Ern
Academic Year   :  2021
Semester        :  1

=============================================================================================================
Course Code:     Marks Obtained:        Grade Obtained:        Remarks:               Quality Point:
=============================================================================================================
BAIT2112         80                     A                      Execellent             8.00
                                                               Total Quality Point:   8.00
```

Figure 4.32 shows that if users want to input the marks for existing year and semester, the marks will be saved at a new created space and compute the new marks.

Task 4:

Part 1: Marks script(marks)

```
#!/bin/bash

marks=$1


if [ $marks -ge 80 -a $marks -le 100 ]
then
   grade="A"
   gpa=4.00
   remarks=" Execellent "
elif [ $marks -ge 75 -a $marks -le 79 ]
then
   grade="A-"
   gpa=3.75
   remarks=" Execellent"
elif [ $marks -ge 70 -a $marks -le 74 ]
then
   grade="B+"
   gpa=3.50
   remarks=" Good"
elif [ $marks -ge 65 -a $marks -le 69 ]
then
   grade="B"
   gpa=3.00
   remarks=" Good "
elif [ $marks -ge 60 -a $marks -le 64 ]
then
   grade="B-"
   gpa=2.75
   remarks=" Pass "
```

```
elif [ $marks -ge 55 -a $marks -le 59 ]
then
   grade="C+"
   gpa=2.5
   remarks=" Pass "
elif [ $marks -ge 50 -a $marks -le 54 ]
then
   grade="C"
   gpa=2.0
   remarks=" Pass "
elif [ $marks -ge 0 -a $marks -le 49 ]
then
   grade="F"
   gpa=0.0
   remarks=" Failed "
elif [ $marks -gt 100 -o $marks -lt 0 ]
then
  grade=" Invalid Marks Range"
fi
```

Part B: Student Validation Form (studV)

```
#!/bin/bash

stud_file="/home/shuern/Downloads/student.txt"

# echo -e "\033[4mSearch Student Details Form\033[0m"
echo "  Student Validation Form"
echo -e "\e[1;36m ------------------------- \e[0m"

 validCode=f
 while [ $validCode = "f" ]
  do
   echo -n " Enter Student ID [00XXX00000]: "; read studID
   echo " "


   studID=${studID^^} #Change to uppercase
   studID=$(echo $studID | tr -d '[[:space:]]' )

   # Check format whether the user has input a value and the format of the input value
   if [ -n $studID ] && [[ $studID =~ ^[0-9][0-9][A-Z][A-Z][Z][0-9][0-9][0-9][0-9][0-9]$ ]]
        then

        # Store the result into a variable
        result="$(echo $studID | grep -i "$studID" "$stud_file")"

```

```
26        # Get value from the above variable
27        while IFS=: read -r programme id name date contact
  mailAdd email
28        do
29         if [[ -n $programme ]]
30          then
31            validCode=t
32             echo "
  ------------------------------------------------------------------
33             echo " "
34
35
36             printf '%-15s%-3s%-17s\n' " Student Name " ":"
  "$name"
37             printf '%-15s%-2s%-s\n' " Programme " ":"
  "$programme"
38
39             echo " "
40
41             echo "
  ------------------------------------------------------------------
42
43         else
44           echo " "
45           echo -e "\033[31;5m This Student ID does not
  exist ! \033[0m"
46           echo " Please enter again."
47           echo " "
48           validCode=f
49         fi
```

```
50          done <<< $result
51    else
52      echo " "
53      echo -e "\033[31;5m Invalid Student ID ! \033[0m"
54      echo " Please enter again."
55      echo " "
56      validCode=f
57    fi
58  done
59
60
61  validChoice=false
62  while [[ $validChoice = "false" ]]
63  do
64   echo " "
65   echo " Press (n) to continue to enter the student's marks: "
66   echo -n " (q) to return to College Management Menu:"; read
    choice
67
68   if ! [[ $choice =~ ^[NnQq]$ ]]
69    then
70      echo " "
71      echo -e "\033[31;5m Invalid Input ! \033[0m"
72      echo " Please enter again."
73      echo " "
74      validChoice=false
75    else
76      validChoice=true
77   fi
78   done
79
80  echo " "
81
82
83
84  case $choice in
85  [Nn]) ./task4b "$studID"
86        ;;
87  [Qq]) ./ManagementMenu
88        ;;
89  esac
90
```

Part 3a : Student Examination MArks Form (task4b)

```
#!/bin/bash

# task4b
# Input the academic year and semester
# create the file and the details
# Check the academic year and sem

course_file="/home/shuern/Downloads/course.txt"
stud_file="/home/shuern/Downloads/student.txt"

new()
{
  studID=$1
  year=$2
  sem=$3
  course_file="/home/shuern/Downloads/course.txt"
  stud_file="/home/shuern/Downloads/student.txt"

  echo " "

     #echo " StudentID: $studID" >> $studID"Result.txt"
     printf '%-17s%-3s%-17s\n' " Student ID " ":" "$studID"
>> $studID"Result.txt"

     result="$(echo $studID | grep -i "$studID" "$stud_file")"

     while IFS=: read -r programme id name date contact
mailAdd email
        do
         #echo " Student Name: $name" >> "$studID"Result.txt
         printf '%-17s%-3s%-17s\n' " Student Name " ":"
"$name" >> $studID"Result.txt"
     done <<< $result
```

```
31
32      #echo " Academic Year: $year" >> "$studID"Result.txt
33      printf '%-17s%-3s%-17s\n' " Academic Year " ":" "$year"
   >> $studID"Result.txt"
34
35      #echo " Semester: $sem" >> "$studID"Result.txt
36      printf '%-17s%-3s%-17s\n' " Semester " ":" "$sem" >>
   $studID"Result.txt"
37
38      echo " " >> "$studID"Result.txt
39
40      echo "
   ============================================================
   >> $studID"Result.txt"
41
42      printf '%-14s%-3s' " Course Code"":" >>
   $studID"Result.txt"
43      printf '%-20s%-3s' " Marks Obtained"":" >>
   $studID"Result.txt"
44      printf '%-20s%-3s' " Grade Obtained"":" >>
   $studID"Result.txt"
45      printf '%-20s%-3s' " Remarks"":" >>  $studID"Result.txt"
46      printf '%-20s%-3s\n' " Quality Point"":" >>
   $studID"Result.txt"
47
48      echo "
   ============================================================
   >> $studID"Result.txt"
49
50 }
51
```

```
52 next() {
53
54
55 validInput=false
56 while [ $validInput = "false" ]
57 do
58  echo " "
59  echo -n " You Want to add new course marks for this semester
   (y)es / (n)o?" ; read respond
60  echo " "
61
62  respond=$(echo $respond | tr "[[:lower:]]" "[[:upper:]]")
63
64  if [[ $respond =~ ^[YN]$ ]]
65  then
66    validInput=true
67   else
68    echo " "
69    echo -e "\033[31;5m Invalid Input ! \033[0m"
70    echo " Please enter again."
71    echo " "
72    validInput=false
73
74   fi
75 done
76
77 }
```

```
 79 studID=$1
 80 echo "  Student Examination Marks Form"
 81 echo -e "\e[1;36m ------------------------------- \e[0m"
 82
 83
 84 continue=true # continue = true
 85 while [[ $continue = "true" ]]
 86  do
 87   echo -n " Academic Year (YYYY): " ; read year
 88
 89   cyear=$(date +%Y) # Current Year
 90
 91   if [[ $year = "" ]]
 92    then
 93     echo " "
 94     echo -e "\033[31;5m Invalid Year ! \033[0m"
 95     echo " Please enter again."
 96     echo " "
 97     continue=true
 98   else
 99    if [[ $(date -d "$year" +"%Y" 2>>/dev/null) ]]
100    then
101     if [ $year -ge 2020 -a $year -le $cyear ] # start from
    acedemic year 2020 and will keep on update the year
102      then
103        continue=false
104     else
105      echo " "
106      echo -e "\033[31;5m Invalid Year ! \033[0m"
107      echo " Please enter again."
108      echo " "
109      continue=true
```

```
110     fi
111     else
112      echo " "
113      echo -e "\033[31;5m Invalid Year ! \033[0m"
114      echo " Please enter again."
115      echo " "
116      continue=true
117   fi
118
119 fi
120
121 done
122
123 while ! [[ $sem =~ ^[1-3]$ ]]
124 do
125  echo -n " Semester (1 / 2 / 3 ): " ; read sem
126
127  if ! [[ $sem =~ ^[1-3]$ ]]
128   then
129     echo -e "\033[31;5m Invalid Input ! \033[0m"
130  fi
131 done
132
133 echo -e "\e[1;34m
    ____________________________________________________________
    \e[0m"
134 echo " "
135
136
137 #-------------------------------------------------------------
138
139
140
```

```
141 if ! [ -f "$studID"Result.txt ] # if file does not exits
142     then
143      touch $studID"Result.txt"     # Create a file for the
    new student
144 else
145   if grep -q -i "$year" $studID"Result.txt" &&  grep -i -w -q
    "$sem" $studID"Result.txt"
146   then
147     echo -e " The result for year \e[1;32m $year \e[0m sem
    \e[1;32m $sem \e[0m is existed."
148
149     next
150     respond=$respond
151
152    if [ $respond = "N" ]
153     then
154       ./studV
155    else [ $respond = "Y" ]
156     validCode=false
157     while [ $validCode = false ]
158     do
159      echo -n " Please enter the course code for checking
    purpose: " ; read courseCode
160
161      courseCode=$(echo $courseCode | tr "[[:lower:]]"
    "[[:upper:]]" )
162
163       if  [ -z $courseCode ] || ! [[ $courseCode =~ ^BA[A-Z]-
    [A-Z][0-9][0-9][0-9][0-9]$ ]]
164       then
165        echo " "
166        echo -e "\033[31;5m Invalid Course Code ! \033[0m"
```

```
167         echo " Please enter a valid Course Code."
168         echo " "
169         validCode=false
170       else validCode=true
171       fi
172     done
173
174       # Check for whether the course code exist
175       if grep -q -w  $courseCode $course_file
176       then
177         #Check duplicate for 1 course
178         # If duplicate
179        if [ -f $studID"Result.txt" ] && grep -q $courseCode
    $studID"Result.txt"
180         then
181          echo " "
182          echo -e "\e[1;34m Mark is added for $courseCode
    \e[0m"
183          echo -e "\033[31;5m Back to Student Validation...
    \033[0m"
184          sleep 1
185          echo " "
186          ./studV
187         fi
188     else  # Code does not exist
189      echo " "
190      echo -e "\033[31;5m Course Code does not exist ! \033[0m"
191      echo " "
192      ./studV
193     fi
194    fi
195  fi
196 fi
197
198
199 echo " "
200 echo " File Name: $studID"Result.txt" "
201  new "$studID" "$year" "$sem"
202
203
204 ./task4c "$studID" "$year" "$sem"
205
```

Part 3b: Student Examination Marks Form (task4c)

```
#!/bin/bash


# input code, marks
# Save the result

studID=$1
year=$2
sem=$3

course_file="/home/shuern/Downloads/course.txt"
stud_file="/home/shuern/Downloads/student.txt"

i=0
sum=0
loop=y
while [ $loop = "y" ]
do

# Once error happen, should directly ask user to prompt or
ask whether they want to continue?

 #--------------
 validCode=false
 while [ $validCode = false ]
 do
  echo -n " Please enter the course code: " ; read courseCode

  courseCode=$(echo $courseCode | tr "[[:lower:]]"
"[[:upper:]]" )


```

```
31   if  [ -z $courseCode ] || ! [[ $courseCode =~ ^BA[A-Z][A-Z]-
[0-9][0-9][0-9][0-9]$ ]]
32   then
33   echo " "
34   echo -e "\033[31;5m Invalid Course Code ! \033[0m"
35   echo " Please enter a valid Course Code."
36   echo " "
37   validCode=false
38 else
39
40 # Check for whether the course code exist
41   if grep -q -w  $courseCode $course_file
42    then
43      validCode=true
44
45      #Check duplicate for 1 course
46      if [ -f $studID"Result.txt" ] && grep -q $courseCode
$studID"Result.txt"
47      then
48         echo " "
49         echo -e "\e[1;34m Mark is added for $courseCode
\e[0m"
50         echo " "
51         validCode=false
52      else
53          validCode=true
54
55       fi
56
57    else  # Code does not exist
58     echo " "
59     echo -e "\033[31;5m Course Code does not exist ! \033[0m"
```

```
60      echo " User is required to create and add the new course
   at the Add New Courses Form."
61      echo " "
62      validCode=false
63      fi
64  fi
65
66  done
67
68  #-----------------------------------------------------
69
70  validMarks=false
71  while [ $validMarks = false ]
72   do
73    echo -n " Enter the marks obtained: " ; read marks
74
75
76    if [[ $marks =~ ^[0-9]+$ ]]
77     then
78       # Run the script file (../ means parents directory)
79
80       . ./marks "$marks"
81
82       if [[ $grade =~ ^[ABCF+-] ]]
83       then
84         validMarks=true
85       else
86        echo " "
87        echo $grade
88        echo -e "\033[31;5m Invalid Marks ! Marks must between
   0 to 100. \033[0m"
89        echo " Please enter a valid marks."
```

```
90          echo " "
91          validMarks=false
92        fi
93    else
94      echo " "
95      echo -e "\033[31;5m Invalid Marks. \033[0m"
96      echo " Please enter again."
97      echo " "
98      validMarks=false
99      fi
100  done
101
102
103 #-------------------------------------------------------------
104 echo " "
105 echo " Saving the student examination marks..."
106 sleep 1
107 echo " "
108
109
110 course_file="/home/shuern/Downloads/course.txt"
111 stud_file="/home/shuern/Downloads/student.txt"
112 marks_file="/home/shuern/Downloads/course.txt"
113
114 # Retrive the credit hours
115 courseDetails="$(echo $courseCode | grep -w "$courseCode"
    $course_file)"
116 IFS=:
117 read -r code name credit <<< $courseDetails
118 credit=$(echo $credit)
119
120 # Calculate quantity point
121 point=`echo $gpa \* $credit | bc`
```

```
124
125       printf '%-18s' " $courseCode" >> $studID"Result.txt"
126       printf '%-22d' " $marks" >> $studID"Result.txt"
127       printf '%-22s' " $grade" >> $studID"Result.txt"
128       printf '%-25s' " $remarks" >>  $studID"Result.txt"
129       printf '%.2f\n' " $point" >> $studID"Result.txt"
130
131 total[i]=$point
132
133
134
135 echo "---------------------------------------"
136
137 #-------------------------------------------------------
138
139 validChoice=false
140 while [ $validChoice = "false" ]
141  do
142   echo " "
143   echo -n " Press (y) to continue to enter the student's
    marks or (b) to back previous or (q) back to Management Menu:
    " ; read choice
144
145   if ! [[ "$choice" =~ ^[YyBbQq]$ ]]
146    then
147      echo " "
148      echo -e "\033[31;5m Invalid Input ! \033[0m"
149      echo " "
150      validChoice=false
151    else
152      validChoice=true
153    fi
154
```

```
156 done
157
158 echo " "
159 echo " "
160
161
162 case $choice in
163 [Yy]) loop=y
164       i=`expr $i + 1`
165       ;;
166 [Bb])for (( x=0; x<=$i; x++ ))
167      do
168       sum=`echo $sum + "${total[x]}" | bc`
169      done
170      printf '%64s%-23s' " " "Total Quality Point:"  >>
    $studID"Result.txt"
171      printf '%.2f\n' $sum >> $studID"Result.txt"
172      echo " " >> $studID"Result.txt"
173      echo " Back to Student Validation Form..."
174      sleep 1
175      echo " "
176      ./studV
177      exit
178      ;;
179 [Qq])for (( x=0; x<=$i; x++ ))
180      do
181       sum=`echo $sum + "${total[x]}" | bc`
182      done
183      printf '%64s%-23s' " " "Total Quality Point:"  >>
    $studID"Result.txt"
184      printf '%.2f\n' $sum >> $studID"Result.txt"
185      echo " " >> $studID"Result.txt"
186      echo " Back to Management Menu..."
187      sleep 1
188      ./ManagementMenu
189      exit
190      ;;
191 esac
192
193 done
```