# Informatics Institute of Technology

Department Of Computing

**Module: ECSI410 – Software Development Principles 01**

**Module Leader: Mr. Guganathan Poravi**

**Coursework**

Date of submission: 03/04/2016

Student ID: w1608490

Student First Name: Chamod Oshan

Student Surname: Hettigodage

## Content



**Figures**



**Table**

# Acknowledgment

My sincere gratitude goes to our module leader Mr.Guhanathan Poravi for providing us with the knowledge and guiding us as we were accomplish this task. My parents for their endless support and encouragement without whom I may not be where I am today.

# Introduction

This gui based java program is created as a Course work of ECSI410 – Software Development Principles 01. This program build to do Awards calculation of a university using student marks. This program designed to get exam marks for each module. After it calculate average of each module and show whether the module is Pass ,Resit or Retake. Then calculate credits in the Leve04,If credits enough to move to the next level, program will be providing the next gui . At last this program will show the degree , class of degree, by concerning about the total credits .

**List of Functional Requirements**

- Login to the program.
- Calculate the Level04 credits and marks.
- Exit with CerHE.
- Calculate the Level05 credits and marks.
- Exit with DipHE.
- Calculate Level06 credits and marks.
- Final Award calculation.

**Non-Functional Requirements**

- Design of gui.
- Number of buttons.
- Font size and font face.
- Switching animations.
- Number of windows.

**Algorithms**

❖ Login to the program.

1) Prompt user for username and password.
2) Get username and password.
3) IF (username=”chamod” AND password=”2015300”)

    Move to the next window

   Else

    Display “Login Failed”

   ENDIF

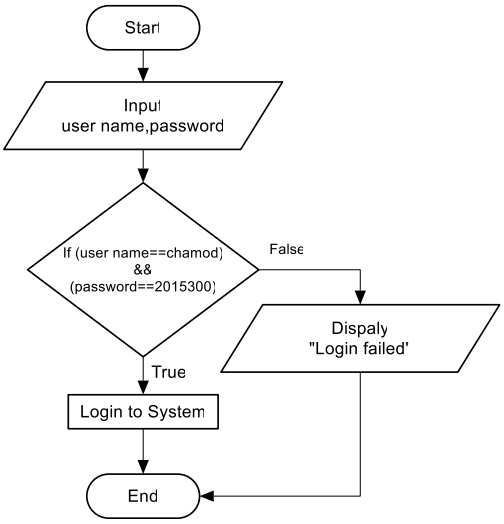



Figure 1.0

Login window code

```
17    //Login Window
18    @FXML
19    private Label lblvalid;
20    @FXML
21    private TextField  txtuser,txtpass;
22
23    public void Login(ActionEvent event)throws Exception{
24        if(txtuser.getText().equals("chamod")&& txtpass.getText().equals("2015300")){
25            Stage primaryStage=new Stage();
26            Parent root =FXMLLoader.load(getClass().getResource("/application/Level04.fxml"));
27            Scene scene = new Scene(root);
28            scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
29            primaryStage.setScene(scene);
30            primaryStage.show();
31        }else{
32            lblvalid.setText("Login Failed");
33        }
34    }
35
36    //Cancel Button
37    @FXML
38    private Button btncancel;
39    @FXML
40    private void  cancelAction(){
41        Stage primaryStage = (Stage) btncancel.getScene().getWindow();
42        primaryStage.close();
43    }
```

## ❖ Calculate the Level04 credits and marks.

1) Prompt user for marks of each ICT exams.
2) Calculate Total marks of each module separately.
3) Calculate Average marks of each module separately.
4) IF(Average>=40)

   Display “pass”

   ELSEIF(Average>=30

   Display “resit”

   ELSE

   Display “retake”

   ENDIF
5) If can give condoned, Display “condoned”.
6) Prompt user for condoned credit.
7) Calculate credits = (Number of pass modules*20+condoned).

8) If get enough credits move to the Level05.

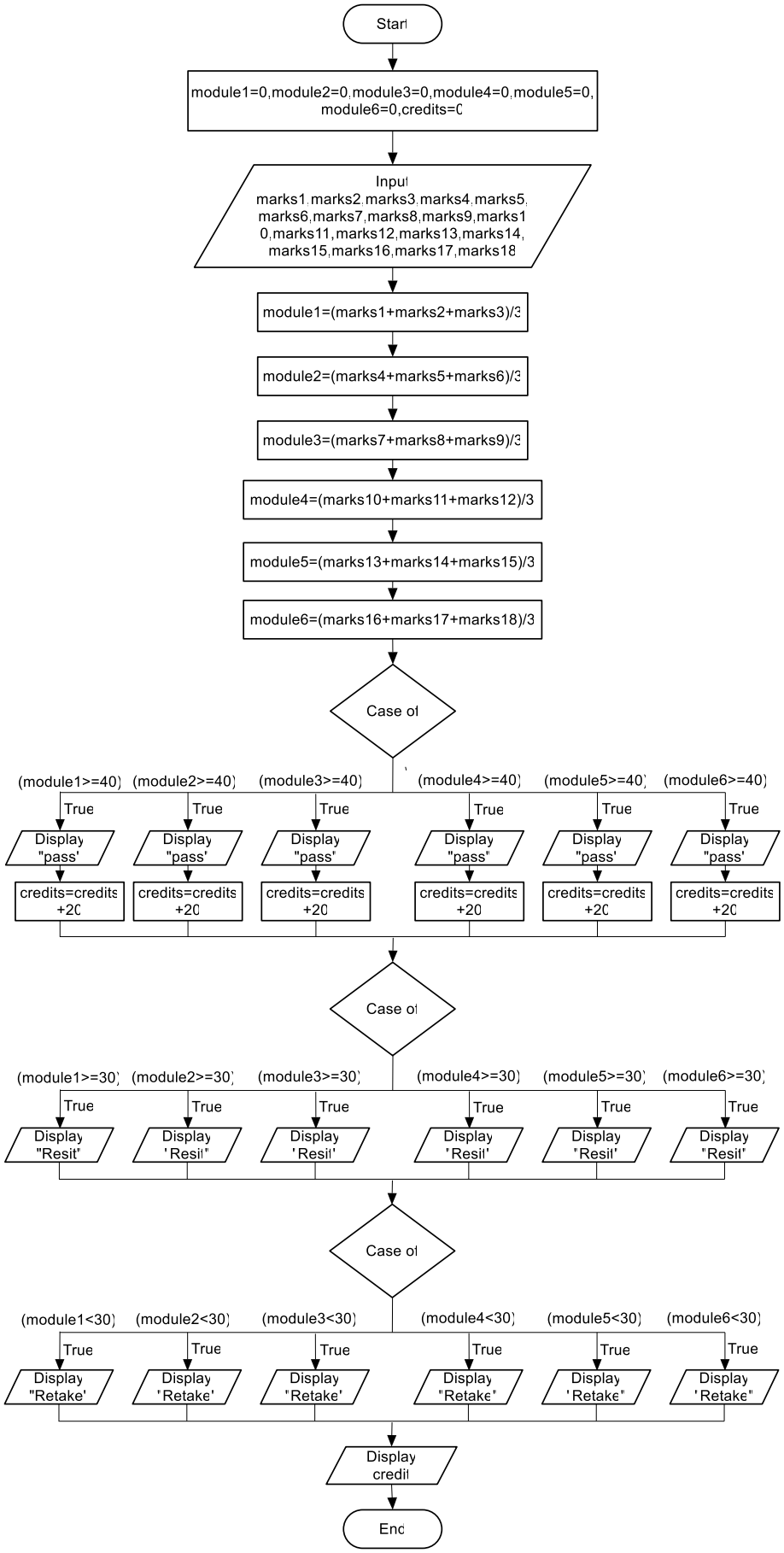



Figure 1.1

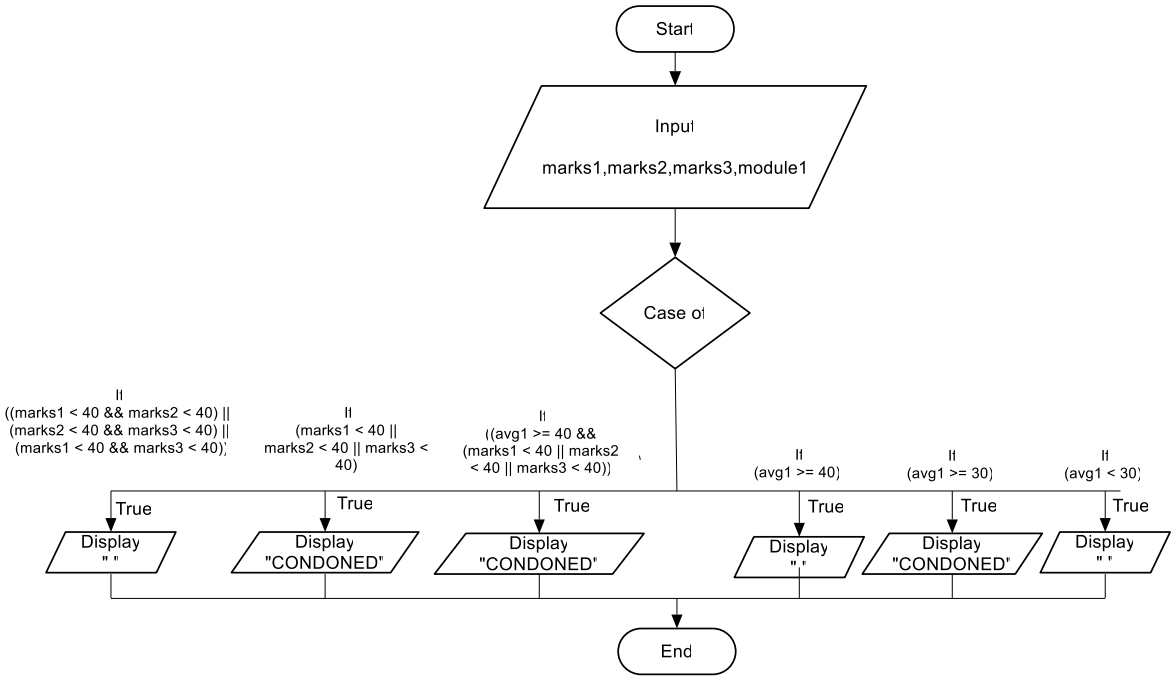



Figure 1.2

```
45      // Level 04 Modules Window
46      @FXML
47      private TextField L4M1Ict01, L4M1Ict02, L4M1Ict03, L4M2Ict01, L4M2Ict02, L4M2Ict03, L4M3Ict01, L4M3Ict02, L4M3Ict03,
48              L4M4Ict01, L4M4Ict02, L4M4Ict03, L4M5Ict01, L4M5Ict02, L4M5Ict03, L4M6Ict01, L4M6Ict02, L4M6Ict03, txtcon;
49      @FXML
50      private Label lblM1, lblM2, lblM3, lblM4, lblM5, lblM6, lblP1, lblP2, lblP3, lblP4, lblP5, lblP6, lblcon1, lblcon2,
51              lblcon3, lblcon4, lblcon5, lblcon6, lblcredit, lblmessage, lblmessage1, lblce;
52
53      public void getinputs(ActionEvent event) {
54          // Module-01
55          int marks1 = Integer.parseInt(L4M1Ict01.getText());
56          int marks2 = Integer.parseInt(L4M1Ict02.getText());
57          int marks3 = Integer.parseInt(L4M1Ict03.getText());
58          int total1 = 0, avg1 = 0;
59          if (marks1<=100&&marks2<=100&&marks3<=100){
60          total1 = (marks1 + marks2 + marks3);
61          avg1 = total1 / 3;
62          lblM1.setText(Integer.toString(avg1));
63          if ((marks1 < 40 && marks2 < 40) || (marks2 < 40 && marks3 < 40) || (marks1 < 40 && marks3 < 40)) {
64              lblcon1.setText("");
65          } else if (marks1 < 40 || marks2 < 40 || marks3 < 40) {
66              lblcon1.setText("CONDONED");
67          }
68          if (avg1 >= 40 && (marks1 < 40 || marks2 < 40 || marks3 < 40)) {
69              lblP1.setText("PASS");
70              lblcon1.setText("CONDONED");
71          } else if (avg1 >= 40) {
72              lblP1.setText("PASS");
73              lblcon1.setText("");
74          } else if (avg1 >= 30) {
75              lblP1.setText("RESIT");
76              lblcon1.setText("CONDONED");
77          } else if (avg1 < 30) {
78              lblP1.setText("RETAKE");
79              lblcon1.setText("");
80          }
81          }else{
82              lblP1.setText("Invalid");
83          }
```

```
        int[] marks = new int[6];
        marks[0] = avg1;
        marks[1] = avg2;
        marks[2] = avg3;
        marks[3] = avg4;
        marks[4] = avg5;
        marks[5] = avg6;
        int credit = 0;
        for (int i = 0; i < marks.length; i++) {
            if (marks[i] >= 30) {
                credit = credit + 20;
                lblce.setText(Integer.toString(credit));
            }

        }

        String[] con = new String[6];
        con[0] = lblcon1.getText();
        con[1] = lblcon2.getText();
        con[2] = lblcon3.getText();
        con[3] = lblcon4.getText();
        con[4] = lblcon5.getText();
        con[5] = lblcon6.getText();
        String CON = "CONDONED";
        for (int j = 0; j < con.length; j++) {
            if (con[j] == CON) {
                credit = credit - 20;
                if (credit>=0){
                lblce.setText(Integer.toString(credit));
                }
            }
        }
```

```
    public void finish(ActionEvent event) {
        int credit = Integer.parseInt(lblce.getText());
        int con = Integer.parseInt(txtcon.getText());
        finalCredit4 = credit + con;
        lblcredit.setText(Integer.toString(finalCredit4));
        if (finalCredit4>120){
            lblmessage.setText("Invalid");
        }else if (finalCredit4 == 120) {
            lblmessage.setText("You are eligible for move to the Level-05");
            lblmessage1.setText("");
        } else if (finalCredit4 == 100) {
            lblmessage.setText("You are eligible for move to the Level-05");
            lblmessage1.setText("but you have to RETAKE your failure Module");
        } else if (finalCredit4 < 100) {
            lblmessage.setText("You are not eligible for move to the Level-05");
            lblmessage1.setText("");
        }

    }
```

```
    public void next1(ActionEvent event)throws Exception{
        if(lblmessage.getText().equals("You are eligible for move to the Level-05")){
        Stage primaryStage=new Stage();
        Parent root =FXMLLoader.load(getClass().getResource("/application/Level05.fxml"));
        Scene scene = new Scene(root);
        scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
        primaryStage.setScene(scene);
        primaryStage.show();
        }
    }
```

- Exit with CerHE.

1) Read Level04 credits.
2) IF(credits=120)

   Display CerHE details.

   ENDIF

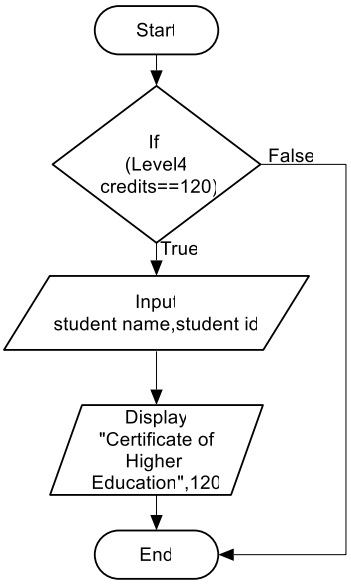



Figure 1.3

```
471     //cerhe
472        @FXML
473          private Label lbldipTopic,lbldiphe1,lbldiphe2,lbldiphe3,lbldiphe4,lbldiphe5,lbldiphe6;
474        @FXML
475          private TextField txtnamediphe,txtiddiphe;
476
477       public void diphe(ActionEvent event)throws Exception{
478           if(L5lblmessage.getText().equals("You are eligible for move to the Level-06")&&
479                   L5lblmessage1.getText().equals("")||
480                   L5lblmessage1.getText().equals("and you passed the RETAKE module of Level-04")){
481           Stage primaryStage=new Stage();
482           Parent root =FXMLLoader.load(getClass().getResource("/application/diphe.fxml"));
483           Scene scene = new Scene(root);
484           scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
485           primaryStage.setScene(scene);
486           primaryStage.show();
487           }
488       }
335    public void Proceed1(ActionEvent event) throws Exception {
336        lblcerTopic.setText("Certificate of Higher Education");
337        lblcerhe1.setText("120");
338        lblcerhe2.setText("--");
339        lblcerhe3.setText("--");
340        lblcerhe4.setText("No Degree");
341        lblcerhe5.setText("--");
342        lblcerhe6.setText("YOU HAVE THE CERTIFICATE OF HIGHER EDUCATION");
343        lblnamecerhe.setText(name);
344        lblidcerhe.setText(Integer.toString(id));
345    }
```

- Calculate the Level05 credits and marks.

1) Prompt user for marks of each ICT exams.
2) Calculate Total marks of each module separately.
3) Calculate Average marks of each module separately.
4) IF(Average>=40)

   Display “pass”

   ELSEIF(Average>=30

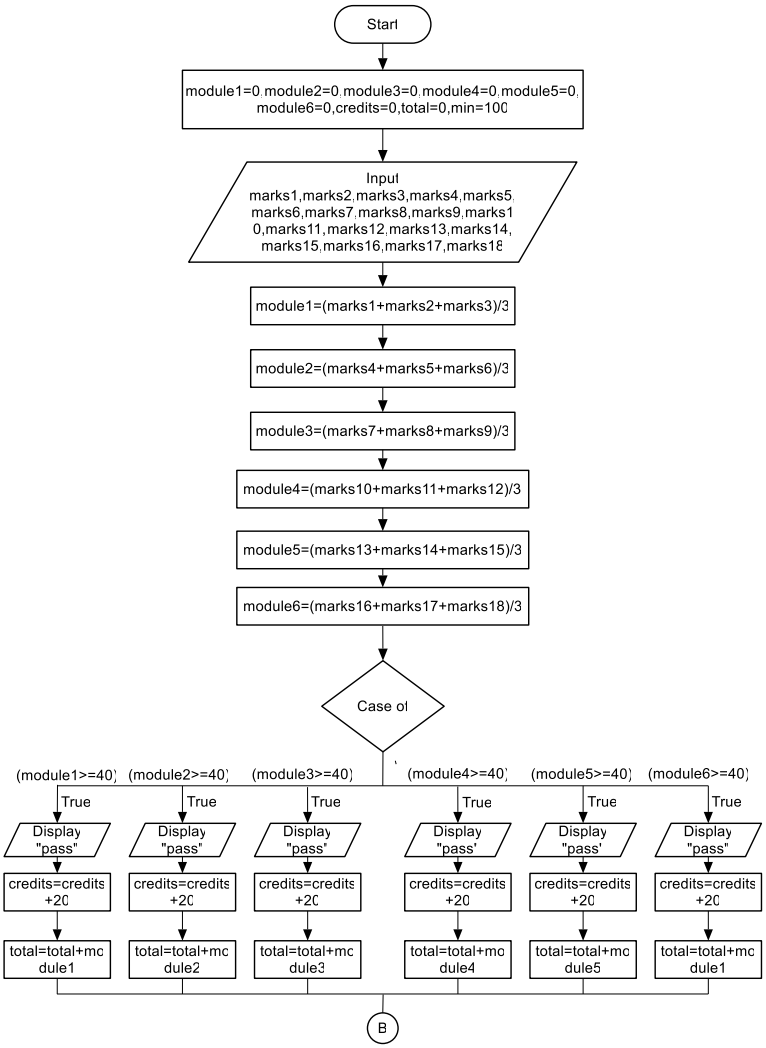
   Display “resit”

   ELSE

   Display “retake”

ENDIF

5) If have a failure module of Level04 Prompt user for retake marks.
6) Calculate retake module credits.
7) Calculate credits = (Number of pass modules*20+retake credits).
8) If get enough credits move to the Level05.



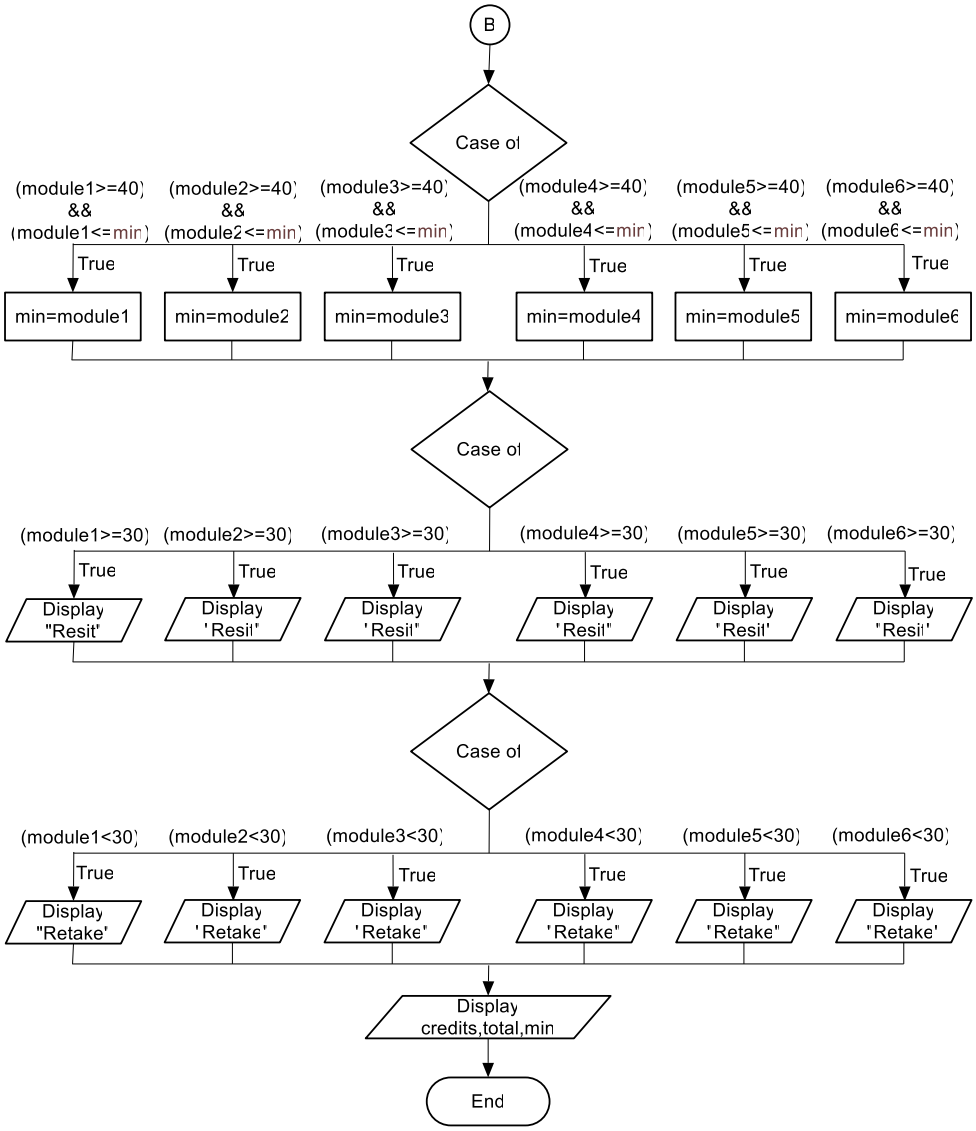



Figure 1.4

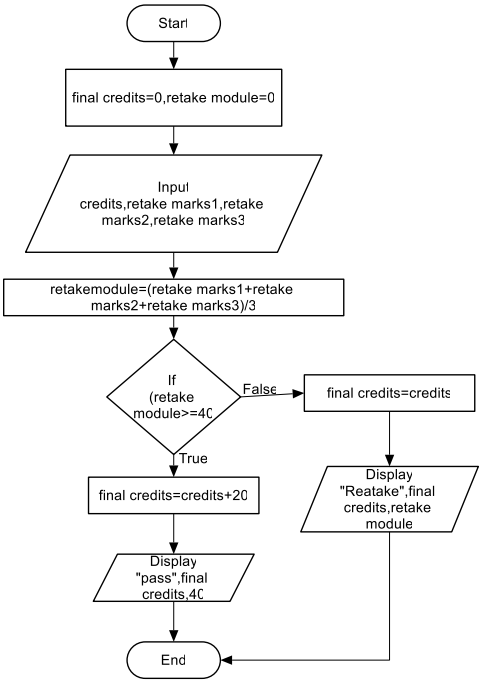



Figure 1.5

```
    // Level 05 Modules Window
    @FXML
    private TextField L5M1Ict01, L5M1Ict02, L5M1Ict03, L5M2Ict01, L5M2Ict02, L5M2Ict03, L5M3Ict01, L5M3Ict02, L5M3Ict03,
            L5M4Ict01, L5M4Ict02, L5M4Ict03, L5M5Ict01, L5M5Ict02, L5M5Ict03, L5M6Ict01, L5M6Ict02, L5M6Ict03, Ict01Re,
            Ict02Re, Ict03Re;

    @FXML
    private Label L5lblM1, L5lblM2, L5lblM3, L5lblM4, L5lblM5, L5lblM6, lblRe, L5lblP1, L5lblP2, L5lblP3, L5lblP4,
            L5lblP5, L5lblP6, lblPRe, L5lblcredit, L5lblmessage, L5lblmessage1, L5lblce, L5lblmin, L5lbltotal;

    public void getinputs1(ActionEvent event) {
        // Module-01
        int marks1 = Integer.parseInt(L5M1Ict01.getText());
        int marks2 = Integer.parseInt(L5M1Ict02.getText());
        int marks3 = Integer.parseInt(L5M1Ict03.getText());
        int total1 = 0, avg1 = 0;
        if (marks1<=100&&marks2<=100&&marks3<=100){
        total1 = (marks1 + marks2 + marks3);
        avg1 = total1 / 3;
        L5lblM1.setText(Integer.toString(avg1));
        if (avg1 >= 40 && (marks1 < 40 || marks2 < 40 || marks3 < 40)) {
            L5lblP1.setText("RESIT");
        }else if (avg1 >= 40) {
            L5lblP1.setText("PASS");
        } else if (avg1 >= 30) {
            L5lblP1.setText("RESITE");
        } else if (avg1 < 30) {
            L5lblP1.setText("RETAKE");
        }
        }else{
            L5lblP1.setText("Invalid");
        }
```

```
        int[] marks = new int[6];
        marks[0] = avg1;
        marks[1] = avg2;
        marks[2] = avg3;
        marks[3] = avg4;
        marks[4] = avg5;
        marks[5] = avg6;
        int credit = 0;
         totalL5 = 0;
        for (int i = 0; i < marks.length; i++) {
            if (marks[i] >= 40) {
                credit = credit + 20;
                L5lblce.setText(Integer.toString(credit));
            }
            if (marks[i] >= 40) {
                if (marks[i] <= min5) {
                    min5 = marks[i];
                    L5lblmin.setText(Integer.toString(min5));
                }
                totalL5 = totalL5 + marks[i];
                L5lbltotal.setText(Integer.toString(totalL5));
            }
        }

    }
```

```
    public void finish1(ActionEvent event) {
        int credit = Integer.parseInt(L5lblce.getText());
        int marks01 = Integer.parseInt(Ict01Re.getText());
        int marks02 = Integer.parseInt(Ict02Re.getText());
        int marks03 = Integer.parseInt(Ict03Re.getText());
        int total0 = 0;// avg0 = 0;
        if (marks01<=100&&marks02<=100&&marks03<=100){
        total0 = (marks01 + marks02 + marks03);
        avg0 = total0 / 3;
        lblRe.setText(Integer.toString(avg0));
        if (avg0 >= 40) {
            lblPRe.setText("PASS");
            lblRe.setText(Integer.toString(40));
        } else if (avg0 >= 30) {
            lblPRe.setText("RESITE");
        } else if (avg0 < 30) {
            lblPRe.setText("RETAKE");
        }
        if (avg0 >= 40) {
            finalCredit5 = credit + 20;
        } else {
            finalCredit5 = credit;
        }
        L5lblcredit.setText(Integer.toString(finalCredit5));
```

```
         if (finalCredit5 == 120 && avg0 < 40) {
             L5lblmessage.setText("You are eligible for move to the Level-06");
             L5lblmessage1.setText("");
         } else if (finalCredit5 == 120) {
             L5lblmessage.setText("You are eligible for move to the Level-06");
             L5lblmessage1.setText("but you have to RETAKE your failure Module");
         } else if (finalCredit5 == 100 && avg0 < 40) {
             L5lblmessage.setText("You are eligible for move to the Level-06");
             L5lblmessage1.setText("but you have to RETAKE your failure Module");
         } else if (finalCredit5 < 100) {
             L5lblmessage.setText("You are not eligible for move to the Level-06");
             L5lblmessage1.setText("");
         } else if (finalCredit5 == 140) {
             L5lblmessage.setText("You are eligible for move to the Level-06");
             L5lblmessage1.setText("and you passed the RETAKE module of Level-04");

         }
         }else{
             lblPRe.setText("Invalid");
         }
     }
```

❖ Exit with DipHE

1) Read Level05 credits.
2) IF(credits=120 OR credits=140)

   Display CerHE details.

   ENDIF

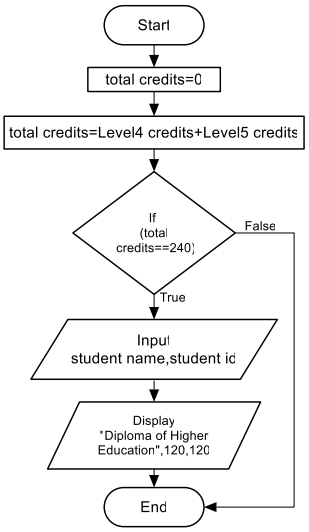



Figure 1.6

```
568     // diphe
569     @FXML
570     private Label lbldipTopic, lbldiphe1, lbldiphe2, lbldiphe3, lbldiphe4, lbldiphe5, lbldiphe6,lblnamediphe,lbliddiphe;
571     public void diphe(ActionEvent event) throws Exception {
572         if (L5lblmessage.getText().equals("You are eligible for move to the Level-06")
573                 && L5lblmessage1.getText().equals("")
574                 || L5lblmessage1.getText().equals("and you passed the RETAKE module of Level-04")) {
575             Stage primaryStage = new Stage();
576             Parent root = FXMLLoader.load(getClass().getResource("/application/diphe.fxml"));
577             Scene scene = new Scene(root);
578             scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
579             primaryStage.setScene(scene);
580             primaryStage.show();
581         }
582     }
583
584     public void Proceed2(ActionEvent event) throws Exception {
585         lbldipTopic.setText("Diploma of Higher Education");
586         lbldiphe1.setText("120");
587         lbldiphe2.setText("120");
588         lbldiphe3.setText("--");
589         lbldiphe4.setText("No Degree");
590         lbldiphe5.setText("--");
591         lbldiphe6.setText("YOU HAVE THE DIPLOMA OF HIGHER EDUCATION");
592         lblnamediphe.setText(name);
593         lbliddiphe.setText(Integer.toString(id));
594     }
```

❖ Calculate Level06 credits and marks.

1) Prompt user for marks of each ICT exams.
2) Calculate Total marks of each module separately.
3) Calculate Average marks of each module separately.
4) IF(Average>=40)

   Display “pass”

   ELSEIF(Average>=30

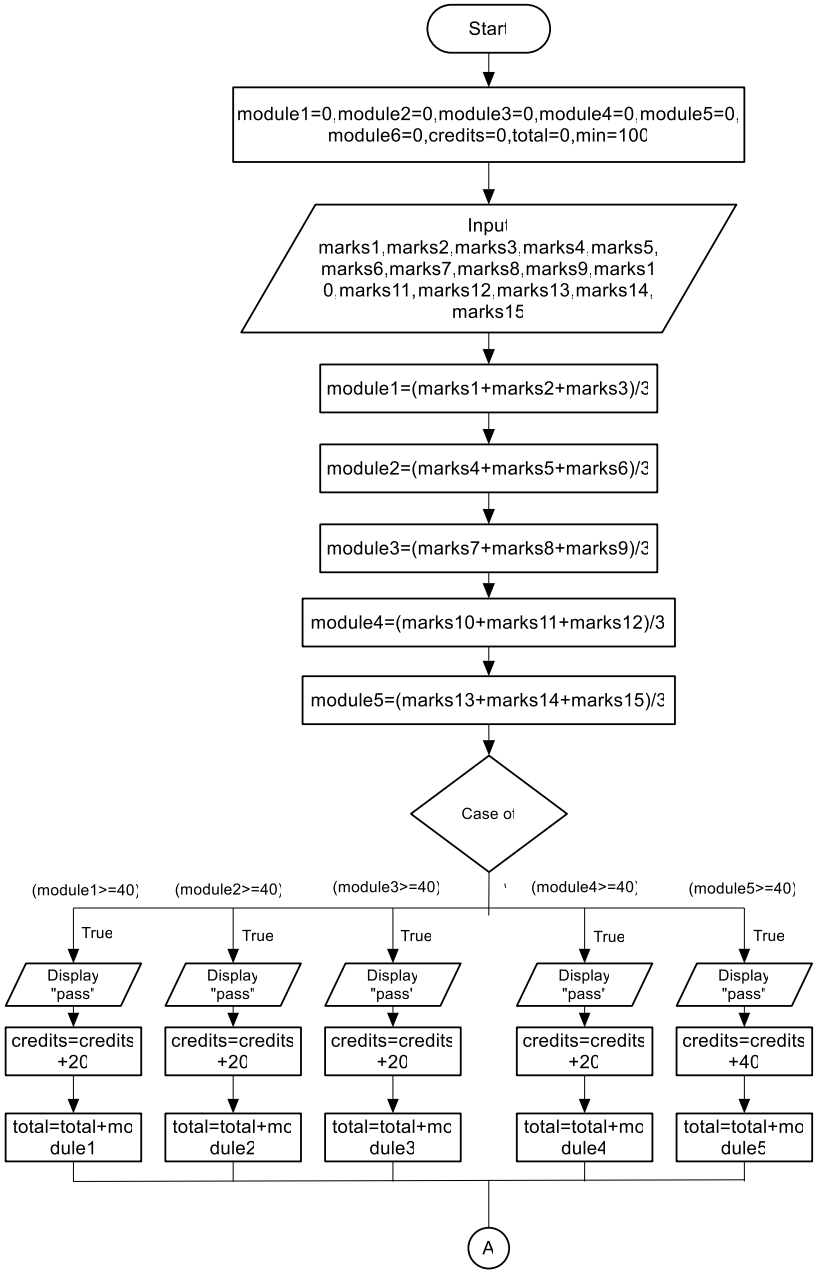
   Display “resit”

   ELSE

Display “retake”

ENDIF

5) If have a failure module of Level04 Prompt user for retake marks.
6) Calculate retake module credits.
7) Calculate credits = (Number of pass single credits modules*20+number of double credits module*40+retake credits).
8) If get enough credits move to the Award calculation.



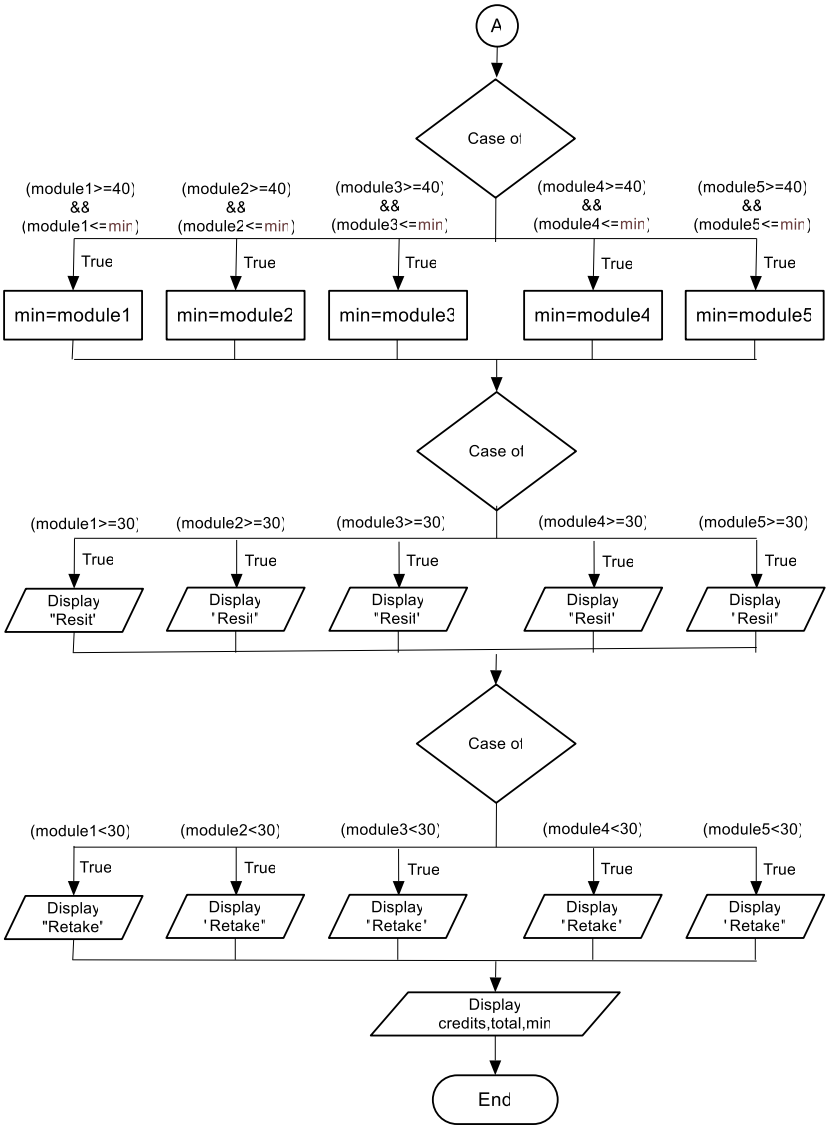



Figure 1.7

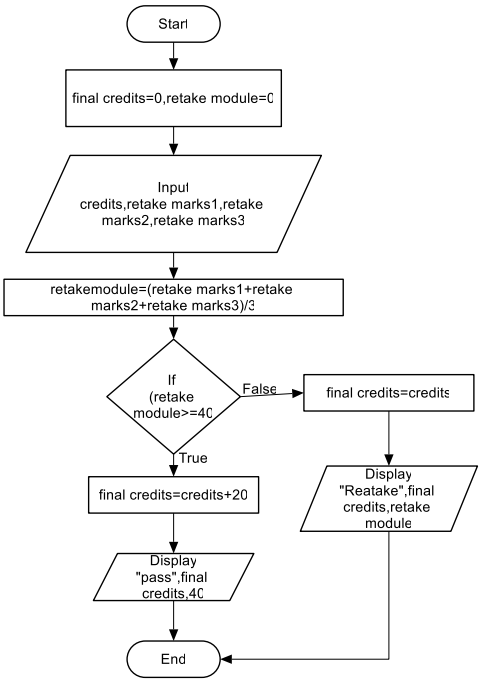



Figure 1.8

```
    // Level 06 Modules Window
    @FXML
    private TextField L6M1Ict01, L6M1Ict02, L6M1Ict03, L6M2Ict01, L6M2Ict02, L6M2Ict03, L6M3Ict01, L6M3Ict02, L6M3Ict03,
            L6M4Ict01, L6M4Ict02, L6M4Ict03, L6M5Ict01, L6M5Ict02, L6M5Ict03, L6M6Ict01, L6M6Ict02, L6M6Ict03,
            L6Ict01Re, L6Ict02Re, L6Ict03Re;

    @FXML
    private Label L6lblM1, L6lblM2, L6lblM3, L6lblM4, L6lblM5, L6lblM6, L6lblRe, L6lblP1, L6lblP2, L6lblP3, L6lblP4,
            L6lblP5, L6lblP6, L6lblPRe, L6lblcredit, L6lblmessage, L6lblmessage1, L6lblce, L6lblmin, L6lbltotal;

    public void getinputs2(ActionEvent event) {
        // Module-01
        int marks1 = Integer.parseInt(L6M1Ict01.getText());
        int marks2 = Integer.parseInt(L6M1Ict02.getText());
        int marks3 = Integer.parseInt(L6M1Ict03.getText());
        int total1 = 0, avg1 = 0;
        if (marks1<=100&&marks2<=100&&marks3<=100){
        total1 = (marks1 + marks2 + marks3);
        avg1 = total1 / 3;
        L6lblM1.setText(Integer.toString(avg1));
        if (avg1 >= 40 && (marks1 < 40 || marks2 < 40 || marks3 < 40)) {
            L6lblP1.setText("RESITE");
        }else if (avg1 >= 40) {
            L6lblP1.setText("PASS");
        } else if (avg1 >= 30) {
            L6lblP1.setText("RESITE");
        } else if (avg1 < 30) {
            L6lblP1.setText("RETAKE");
        }
        }else{
            L6lblP1.setText("Invalid");
        }

         int[] marks = new int[4];
         marks[0] = avg1;
         marks[1] = avg2;
         marks[2] = avg3;
         marks[3] = avg4;

         int credit = 0;
         totalL6 = 0;
         for (int i = 0; i < marks.length; i++) {
             if (marks[i] >= 40) {
                 credit = credit + 20;
                 L6lblce.setText(Integer.toString(credit));

             }
             if (marks[i] >= 40) {
                 if (marks[i] <= min6) {
                     min6 = marks[i];
                     L6lblmin.setText(Integer.toString(min6));
                 }

                 totalL6 = totalL6 + marks[i];
                 L6lbltotal.setText(Integer.toString(totalL6));
             }
         }
         if (avg5 >= 40) {
             credit = credit + 40;
             L6lblce.setText(Integer.toString(credit));
         }
         if (avg5 <= min6 && avg5 >= 40) {
             min6 = avg5;
             L6lblmin.setText(Integer.toString(min6));
         }
         if (avg5 >= 30) {
             totalL6 = totalL6 + avg5;
             L6lbltotal.setText(Integer.toString(totalL6));
         }

     }

     public void finish2(ActionEvent event) {
         int credit = Integer.parseInt(L6lblce.getText());
         int marks001 = Integer.parseInt(L6Ict01Re.getText());
         int marks002 = Integer.parseInt(L6Ict02Re.getText());
         int marks003 = Integer.parseInt(L6Ict03Re.getText());
         int total00 = 0;// avg00 = 0;
         if (marks001<=100&&marks002<=100&&marks003<=100){
         total00 = (marks001 + marks002 + marks003);
         avg00 = total00 / 3;
         L6lblRe.setText(Integer.toString(avg00));
         if (avg00 >= 40) {
             L6lblPRe.setText("PASS");
             L6lblRe.setText(Integer.toString(40));
         } else if (avg00 >= 30) {
             L6lblPRe.setText("RESITE");
         } else if (avg00 < 30) {
             L6lblPRe.setText("RETAKE");
         }
         if (avg00 >= 40) {
             finalCredit6 = credit + 20;
         } else {
             finalCredit6 = credit;
         }
         L6lblcredit.setText(Integer.toString(finalCredit6));
```

```
        if (finalCredit6 == 120 && avg00 < 40) {
            L6lblmessage.setText("You are eligible for the HONORS DEGREE");
            L6lblmessage1.setText("");
        } else if (finalCredit6 == 120) {
            L6lblmessage.setText("You are eligible for the HONORS DEGREE");
            L6lblmessage1.setText("nd you passed the RETAKE module of Level-05");
        } else if (finalCredit6 >= 60 && avg00 < 40) {
            L6lblmessage.setText("You are eligible for the NON-HONORS DEGREE");
            L6lblmessage1.setText("you can RETAKE your failure modules");
        } else if (finalCredit6 < 60) {
            L6lblmessage.setText("You have not passed the Level-06");
            L6lblmessage1.setText("");
        } else if (finalCredit6 == 140) {
            L6lblmessage.setText("You are eligible for the HONORS DEGREE");
            L6lblmessage1.setText("and you passed the RETAKE module of Level-05 ");

        }
        }else{
            L6lblPRe.setText("Invalid");
        }
    }
```

```
    public void next3(ActionEvent event) throws Exception {
        if (L6lblmessage.getText().equals("You are eligible for the HONOURS DEGREE")
                || L6lblmessage.getText().equals("You are eligible for the NON-HONOURS DEGREE")) {
            Stage primaryStage = new Stage();
            Parent root = FXMLLoader.load(getClass().getResource("/application/Award.fxml"));
            Scene scene = new Scene(root);
            scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
            primaryStage.setScene(scene);
            primaryStage.show();
        }
    }
```

❖ Final Award calculation

1) Read final credits of each Level ,total, average, retake marks.
2) Calculate total credits
3) Calculate total marks.
4) Reduce the minimum module marks.
5) Calculate the average.
6) IF(Credits>360)

    Display"Invalid"

ELSEIF(Credits=360)

    Display "B.Sc.Honors"

    IF(average>=70)

        Display"1st class Honors"

    ELSEIF(average>=60)

        Display"2nd class upper division"

    ELSEIF(average>=50)

        Display"2nd class lower division"

    ELSEIF(average>=40)

        Display"3rd class Honors"

ELSEIF(Credits>=300 AND total<360)

    Display"B.Sc"

ENDIF

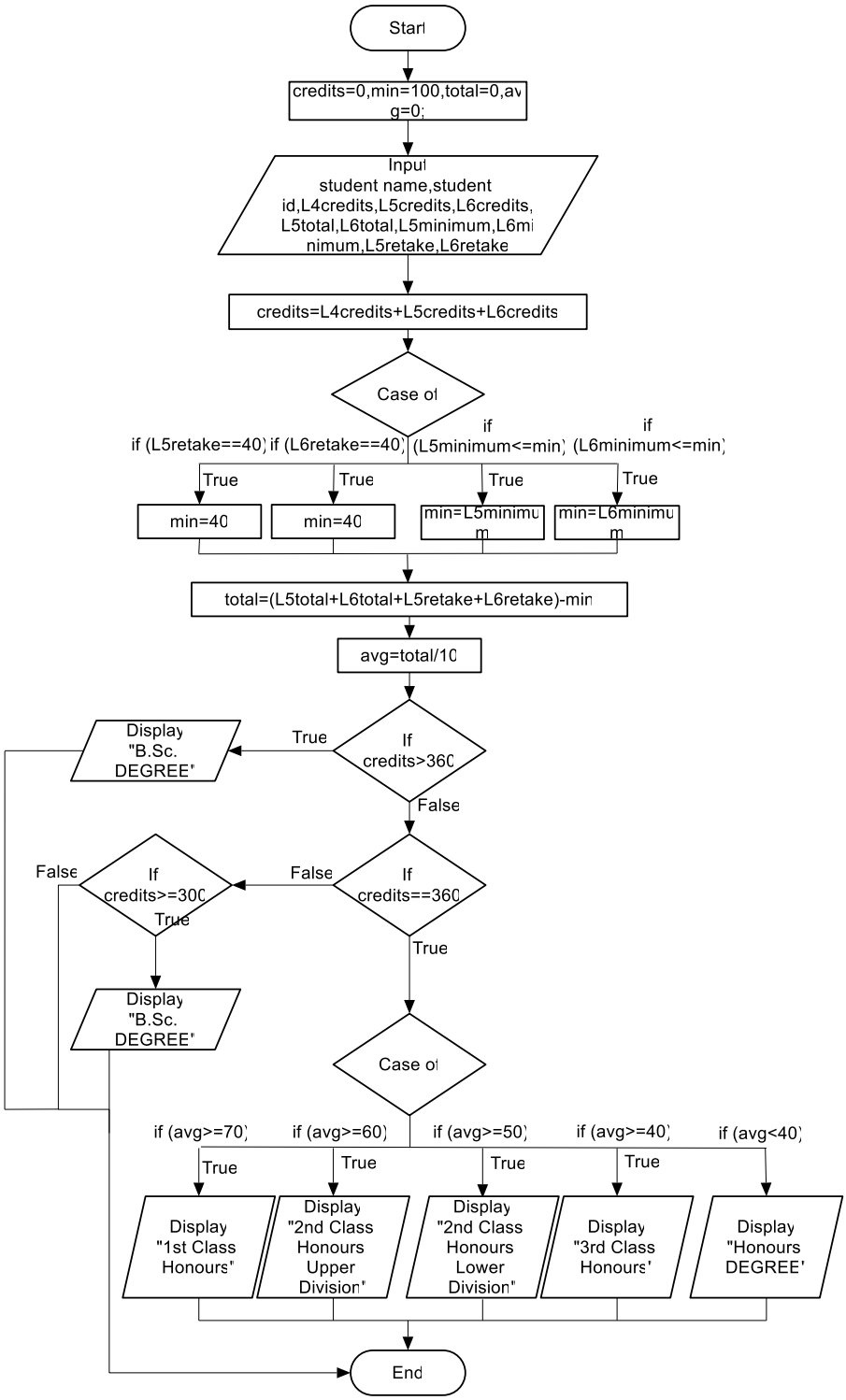

Start
credits=0,min=100,total=0,avg=0;
Input
student name,student id,L4credits,L5credits,L6credits,L5total,L6total,L5minimum,L6minimum,L5retake,L6retake
credits=L4credits+L5credits+L6credits
Case of
if (L5retake==40)
if (L6retake==40)
if (L5minimum<=min)
if (L6minimum<=min)
True
True
True
True
min=40
min=40
min=L5minimum
min=L6minimum
total=(L5total+L6total+L5retake+L6retake)-min
avg=total/10
If credits>360
True
Display "B.Sc. DEGREE"
False
If credits==360
False
If credits>=300
False
True
True
Display "B.Sc. DEGREE"
Case of
if (avg>=70)
if (avg>=60)
if (avg>=50)
if (avg>=40)
if (avg<40)
True
True
True
True
Display "1st Class Honours"
Display "2nd Class Honours Upper Division"
Display "2nd Class Honours Lower Division"
Display "3rd Class Honours"
Display "Honours DEGREE"
End

Figure 1.9

```
    @FXML
    private Label lblawName,lblawId,lblawcredit, lblawavg,lblawL4c,lblawL5c,lblawL6c,lblawL5t,lblawL6t,lblawL5m,lblawL6m,
    lblawL5r,lblawL6r,lblawmin, lblaw1, lblaw2, lblaw3;

    public void proceed(ActionEvent event) throws Exception {
        if (avg0 >= 40) {
            avgRe1=40;
        }else{
            avgRe1=0;
        }
        if (avg00 >= 40) {
            avgRe2=40;
        }else{
            avgRe2=0;
        }
        lblawName.setText(name);
        lblawId.setText(Integer.toString(id));
        lblawL4c.setText(Integer.toString(finalCredit4));
        lblawL5c.setText(Integer.toString(finalCredit5));
        lblawL6c.setText(Integer.toString(finalCredit6));
        lblawL5t.setText(Integer.toString(totalL5));
        lblawL6t.setText(Integer.toString(totalL6));
        lblawL5m.setText(Integer.toString(min5));
        lblawL6m.setText(Integer.toString(min6));
        lblawL5r.setText(Integer.toString(avgRe1));
        lblawL6r.setText(Integer.toString(avgRe2));

        int credits = 0, min = 100, total = 0, avg = 0;
        credits = finalCredit4 + finalCredit5 + finalCredit6;
        lblawcredit.setText(Integer.toString(credits));
        int[] marks = new int[2];
        marks[0] = min5;
        marks[1] = min6;
        if (avgRe1 == 40) {
            min = 40;
            lblawmin.setText(Integer.toString(min));
        }


        if (avgRe2 == 40) {
            min = 40;
            lblawmin.setText(Integer.toString(min));
        }

        for (int i = 0; i < marks.length; i++) {
            if (marks[i] <= min) {
                min = marks[i];
                lblawmin.setText(Integer.toString(min));
            }
        }
        total = (totalL5 + totalL6 + avgRe1 + avgRe2) - min;
        avg = total / 10;
        lblawavg.setText(Integer.toString(avg));
        if (credits > 360) {
            lblaw1.setText("Invalid");
            lblaw2.setText("Invalid");
            lblaw3.setText("Invalid");
        }else if (credits == 360) {
                  lblaw1.setText("B.Sc. Honors");
                  if (avg >= 70) {
                     lblaw2.setText("1st Class Honors");
                     lblaw3.setText("B.Sc. 1st Class Honors DEGREE");
                  } else if (avg >= 60) {
                     lblaw2.setText("2nd Class Honors Upper Division");
                     lblaw3.setText("B.Sc. 2nd Class Honors Upper Division DEGREE");
                  } else if (avg >= 50) {
                     lblaw2.setText("2nd Class Honors Lower Division");
                     lblaw3.setText("B.Sc. 2nd Class Honors Lower Division DEGREE");
                  } else if (avg >= 40) {
                     lblaw2.setText("3rd Class Honors");
                     lblaw3.setText("B.Sc. 3rd Class Honors DEGREE");
                  } else {
                     lblaw2.setText("No Class");
                     lblaw3.setText("B.Sc. Honors DEGREE");
                  }
        } else if (credits >= 300) {
            lblaw1.setText("B.Sc.");
            lblaw3.setText("B.Sc. DEGREE");
            lblaw2.setText("--");
        }
```

# Test Cases

## Black-Box Testing

| INPUT | EXPECTED RESULT | ACTUAL RESULT |
|---|---|---|
| **Enter Marks For Each Module** | It will be able to enter only Integers. | It will be able to enter only Integers. |
| **Press Ok** | Calculate average of each module and print whether Pass, Resit or Retake and Print current credits. | Calculate average of each module and print whether Pass, Resit or Retake and Print current credits. |

| | | |
|---|---|---|
| **Enter Condoned** | If module has condoned enter condoned credits. Else enter 0. | If module has condoned enter condoned credits. Else enter 0. |
| **Press Finish** | Calculate final credits, adding condoned and display states. | Calculate final credits, adding condoned and display states. |
| **Press CerHE/DipHE** | Exit for the CertHE/DipHE | Exit for the CertHEDipHE. |
| **Press Next Level** | Move to the next level | Move to the next level |
| **Press Ok Button in the Level05 and Level06** | Calculate average of each module and print whether<br>Pass, Resit or Retake and<br>Print current credits.<br>Calculate total of averages and minimum average, | Calculate average of each module and print whether<br>Pass, Resit or Retake and<br>Print current credits.<br>Calculate total of averages and minimum average, |
| **Press Finish Button in the Level05 and Level06.** | Calculate final credits, adding Retake module credits and display states. | Calculate final credits, adding Retake module credits and display states. |
| **Press Proceed** | Calculate final Award | -. |

Table 1.0

| INPUT | EXPECTED RESULT | ACTUAL RESULT |
|---|---|---|
| **45 855 799** | invalid | invalid |
| **45 85 79** | Average = 69% , Pass | Average = 69% , Pass |
| **80 80 20** | Average = 60% ,L4- Pass, con L5/L6-Resit | Average = 60% ,L4- Pass, con L5/L6-Resit |
| **10 10 10** | Average = 10% ,Retake | Average = 10% ,Retake |
| **40 40 40** | Average = 40% , Pass | Average = 40% , Pass |
| **39 39 39** | Average = 39% , L4-Resit, con L5/L6-Resit | Average = 39% , L4-Resit, con L5/L6-Resit |
| **100 100 100** | Average = 100% , Pass | Average = 100% , Pass |
| **Retake L04:-85 99 78** | Average = 40% , L5/L6-Pass | Average = 40% , L5/L6-Pass |
| **45 85 79**<br>**80 80 20**<br>**10 10 10**<br>**40 40 40**<br>**39 39 39**<br>**100 100 100** | L4-Credits=60, Con=40,Finla credits=100<br>L5-Total = 269, Minimum = 40, Credits=80, Final credits=100<br>L6- Total = 208, Minimum = 40, Credits=60, Final credits=80 | L4-Credits=60, Con=40,Finla credits=100<br>L5-Total = 269, Minimum = 40, Credits=80, Final credits=100<br>L6- Total = 208, Minimum = 40, Credits=60, Final credits=80 |
| **Credits=L4-100 L5-120 L6-140**<br>**Total=L5-346 L6-296**<br>**Minimum=L5-40 L6-47**<br>**Retake=L5-40 L6-40** | Credits=360, Average=68, Minimum=40,Degree=B.Sc.Honors, Degree Class=2nd class Honors upper division,<br>Display "B.Sc 2nd class Honors upper division<br>DEGREE. | Credits=360, Average=68, Minimum=40,Degree=B.Sc.Honors, Degree Class=2nd class Honors upper division,<br>Display "B.Sc 2nd class Honors upper division<br>DEGREE. |

Table 1.1

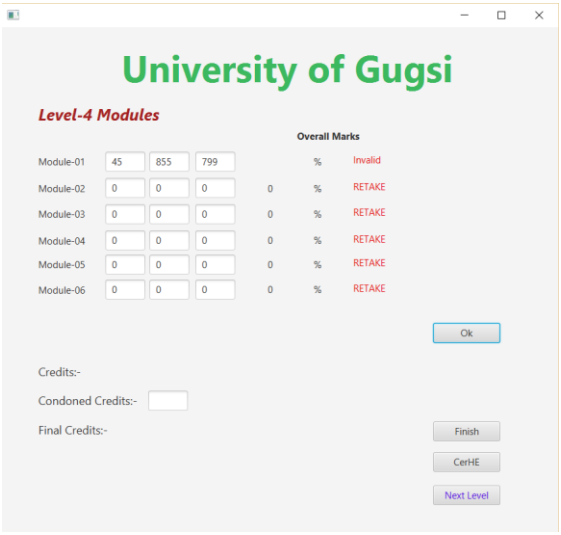



Figure 2.0

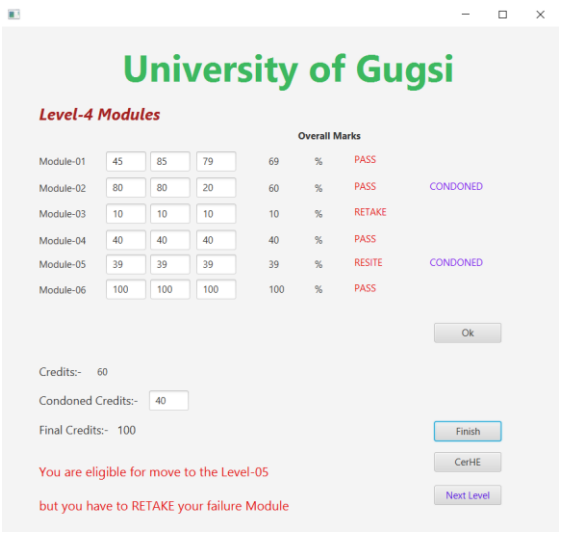



Figure 2.1

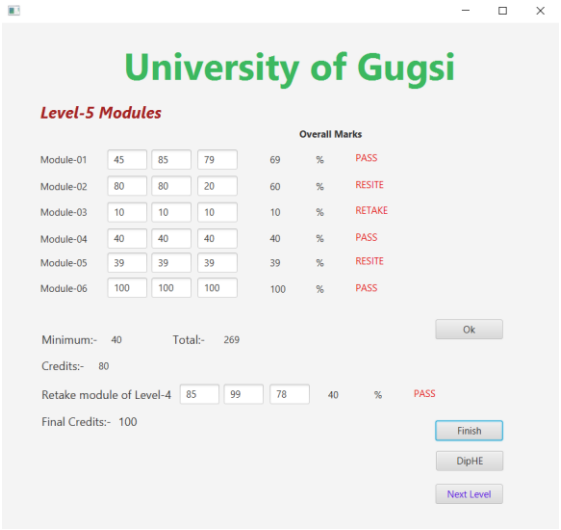



Figure 2.2

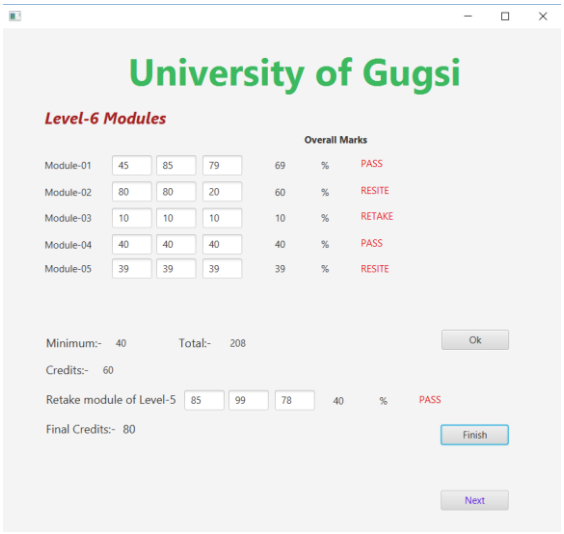



Figure 2.3

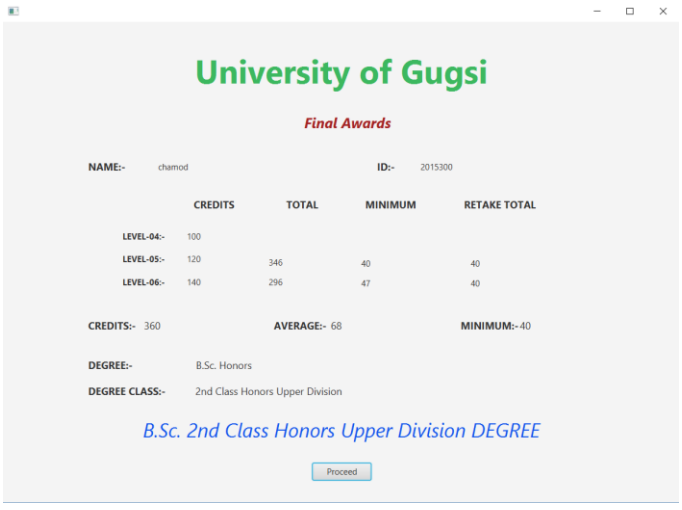



Figure 2.5

## White Box Testing

| CONDITION | YES | NO |
|---|---|---|
| If user name==”chamod”&& password==”2015300” | Login to the program. | “Login failed “error shown . |
| If (marks1<=100&&marks2<=100&& marks3<=100) | Move to the next condition. | Display “invalid” |
| If ((marks1 < 40 && marks2 < 40) \|\| (marks2 < 40 && marks3 < 40) \|\| (marks1 < 40 && marks3 < 40)) | Display “” | Move to the next condition. |
| If(marks1<40\|\|marks2<40 \|\|marks3<40) | Display “Condoned” | Move to the next condition. |
| If ( module average>=40 &&(marks1<40\|\|marks2<40 \|\|marks3<40)) | Display “pass” , “Condoned” | Move to the next condition. |
| If module average>=40 | Display “Pass” | Move to the next condition. |
| If module average>=30 | Display “Resit” and “condoned” | Move to the next condition. |
| If module average<30 | Display “Retake” | -. |
| If module average>=30 | Credits= Credits+20 | Move to the next condition. |
| If final credits==120 | Can move to the next level. | Can’t move to the next level. |
| If final credits==100 | Can move to the next level. Display “you have to Retake your failure module” | Can’t move to the next level |
| If final credits<100 | Can’t move to the next level. | -. |
| If total credits==360 | Display ”Honors degree” | Move to the next condition. |
| If total average marks>=70 | Display “1st class honors degree” | Move to the next condition. |
| If total average marks>=60 | Display “2nd class honors upper division degree” | Move to the next condition. |
| If total average marks>=50 | Display “2nd class honors lower division degree” | Move to the next condition. |
| If total average marks>=40 | Display “3rd class honors degree” | Display “No class ” |

Table 1.2

## Conclusion

I learnt what is the javaFX and I knew about how to design a GUI based program. I got some knowledge about how to do a course work properly.