Islington College

Programming CS40001NI

Coursework 1

Submitted By:

Abiral Joshi-Shrestha

ID: NP01NT4A170008

Group: L1N1

Date: 19th Jan'18

Submitted To:

Mr. Bhim Bahadur Sunar

Lecturer, IT

Contents

1 Introduction

JAVA is a programming language, its target is to write a program once and then run that program on multiple operating systems. The JAVA program in this project was created using Blue J.

The program is an application which keeps track of the activities of a given educational organisation. The program was created to help manage the records and information of the organisation. The application stores the information about particular student such as their name, roll number, etc. keeps track of the classes and courses conducted, and also regarding the fee structure.

This program can be used by any educational organisation or institutes to manage the information required by them to maintain the proper functioning of their daily activities.

While doing this project, I learnt many things about JAVA programming language. The study and work I did while doing this coursework helped me better understand JAVA, cleared my confusions and misunderstandings. I also learnt how to make programs compact and yet, effective. The knowledge I gained while doing this prject can be very helpful in my future career as a programmer.

2 Class Diagram

2.1 Course (Parent Class)

Course
All instance variables + courseName : String + instructorName : String + studentName : String + totalHours : int
All methods + Course (String courseName, String instructorName, int totalHours) + courseName() : String + studentName() : String + display() : void + totalHours() : int + instructorName() : String

2.2 Certification (Child Class)

Certification
All instance variables - examDate : String - startDate : String - examCenter : String - start : boolean - validTill : String - certificateAwardedBy : String - courseFee : int
All methods + Certification (String courseName, String instructorName, int totalHour, int courseFee, String certificateAwardedBy, String validTill) + examDate() : String + startDate() : String + start() : Boolean + validTill() : String + certificateAwardedBy() : String + examCenter() : String . .

2.3 Professional (Child Class)

Professional
All instance variables - enrolDate : String - start : boolean - complete : boolean - roomNo : String - dailyHour : int - downPayment : int
All methods + Professional (String courseName, String instructorName, int courseFee, int totalHours) + enrolStudent (String studentName, String enrolDate, int downPayment, String roomNo) : void + enrolDate() : String + roomNo() : String + courseFee() : int + dailyHour() : int +downPayment() : int + start() : boolean + complete() : boolean + courseFee(int courseFee) : void + dailyHour(int dailyHour) : void

3 Pseudo Code

The pseudo code for this program are;

A) Course (Parent Class)

```
Call Course( String courseName, String instructorName, int totalHours)
        DO
                studentName = " ";
                this.courseName = courseName;
                this.instructorName = instructorName;
                this.totalHours = totalHours
        END DO
Call String getcourseName()
        DO
                Return courseName;
        END DO
Call String getinstructorName()
        DO
                Return instructorName;
        END DO
Call String studentName()
        DO
                Return studentName;
        END DO
Call String gettotalHours()
        DO
                Return totalHours;
        END DO
Call String setstudentName(Stringstudent)
        DO
                this.studentName = studentName;
        END DO
Call void display()
        DO
                IF (studentName.lenght()==0)
```

```
                DO
                DISPLAY("The name of course is " +courseName)
                DISPLAY("The instructor's name is "
        +instructorName)
                DISPLAY("The total hours taken is " +totalHours)
                END DO
        ELSE
                DO
                DISPLAY("The name of course is " +courseName)
                DISPLAY("The name of the instructor is "
+instructorName     )
                DISPLAY("Total hours taken is " +totalHours)
                DISPLAY("Student's name is " +studentName)
                END DO
END DO
```

B) Certification (Child Class)

Call Certification (String courseName, String instructorName, int totalHour, int courseFee, String certificateAwardedBy, String validTill)

DO

Super (courseName, instructorName, totalHours);

this.courseFee = courseFee;

this.certificateAwardedBy = certificateAwardedBy;

this.validTill = validTill;

startDate = “”;

examDate = “” ;

examCenter = “” ;

start = false;

END DO

Call int getcourseFee()

DO

Return courseFee;

END DO

Call String getexamDate()

DO

Return examDate;

END DO

Call String startDate()

DO

Return startDate;

END DO

Call String getexamCenter()

DO

Return examCenter;

END DO

Call String gecertificateAwardedBy()

DO

Return certificateAwardedBy;

```
        END DO
    Call String getvalidTill()
        DO
            Return validTill;
        END DO
    Call boolean getstart()
        DO
            Return start;
        END DO
    Call void setcourseFee(int courseFee)
        DO
            If (start==falce)
                DO
                    this.courseFee = courseFee;
                ELSE DO
                    DISPLAY("The fee is fixed.")
                END DO
Call void enrolStudent( String studentName, String startDate, String
examDate, String examCenter)
    DO
        IF(start==false)
            DO
                Super.setstudentName(studentName);
                start=true;
            END DO
        ELSE
            DO
                DISPLAY("The course has already started.")
            END DO
        this.startDate = startDate;
        this.examDate = examDate;
        this.examCenter = examCenter;
    END DO
Call void display()
```

```
DO
    Super.display()
    If(start==true)
        DO
            System.out.println("The course starts from "
            +startDate);
            System.out.println("The student's name is "
            +studentName)
            System.out.println(" Exam date is " +examDate);
            System.out.println(" Exam center is "
            +examCenter);
            System.out.println("Certificate awarded by "
            +certificateAwardedBy);
            System.out.println(" Validity till " +validTill);
        END DO
END DO
```

C) Professional (Child Class)

```
Call Professional( String courseName, String instructorName, int
courseFee, int totalHours, int dailyHours)
    DO
        Super(courseName, instructorName, totalHours);
        this.courseFee = courseFee;
        this.dailyHour = dailyHour;
        downPayment = 0;
        enrolDate = “”;
        roomNo = “”;
        start = false;
        complete = false;
    END DO
Call String getenrolDate()
    DO
        Return enrolDate;
    END DO
Call String getroomNo()
    DO
        Return roomNo;
    END DO
Call String getcourseFee()
    DO
        Return courseFee;
    END DO
Call int getdailyHour()
    DO
        Return dailyHour;
    END DO
Call int downPayment()
    DO
        Return downPayment;
    END DO
```

```
Call boolean getstart()
        DO
                Return start;
        END DO
Call boolean getcomplete()
        DO
                Return complete;
        END DO
Call void setcourseFee(int courseFee)
        DO
                this.courseFee = courseFee;
        END DO

Call void setdailyHour(int dailyHour)
        DO
                this.dailyHour = dailyHour;
        END DO
Call void enrolStudent( String studentName, String enrolDate, int
downPayment, String roomNo)
        DO
                If(complete = true)
                        DO
                                DISPLAY(“Course has been completed.”);
                        END DO
                ELSE
                        DO
                                Super.studentName(“ “);
                                enrolDate = “”;
                                roomNo = “”;
                                downPayment = 0;
                                start = false;
                                complete = true;
                        END DO
        END DO
```

```
Call void print ( String courseName, String instructorName, int
courseFee)
        DO
                Super.getcourseName();
                DISPLAY(“Name of course is ” +courseName)
                DISPLAY(“Enrol date is ” +enrolDate)
                DISPLAY(“ Down payment is ” +downPayment)
        END DO
Call void display()
        DO
                If(start = true)
                        DO
                                DISPLAY(“Course is pending.”)
                                DISPLAY(“Enrol date is ” +enrolDate)
                                DISPLAY(“Down payment is ”
                                +downPayment)
                        END DO
```

4 Method Description

4.1 Course (Parent Class)

---

getcourseName

This method is used to get the Student Name.

getinstructorName

This method is used to get the Instructor Name.

getstudentName

This method is used to get student name.

gettotalHours

This method is used to get the total hours.

setstudentName

This method is used to set student name.

display

This method is used to display the course name, instructor's name, total hours and student's name.

4.2 Certification (Child Class)

getcourseFee

This method is used to get the course fee.

getexamDate

This method is used to get the exam date.

getstartDate

This method is used to get the start of the exam.

getexamCenter

This method is used to get the center of the exam.

getcertificateAwardedBy

This method is used to get who the certificate is awarded by.

getvalidTill

This method is used to get the validity period of the certificate.

getstart

This method is used to get the value of start.

4.3 Professional (Child Class)

getenrolDate

This method is used to get the enrol date.

getroomNo

This method is used to get the room number.

getcourseFee

This method is used to get the course fee.

getdailyHour

This method is used to get the daily hours.

getdownPayment

This method is used to get the down payment.

getstart

This method is used to confirm if the course is started or not.

getComplete

This method is used to confirm the end of the course.

setcourseFee

This method is used to set the new value to the courseFee.

setdailyHours

This method is used to set the daily hours of the classes.

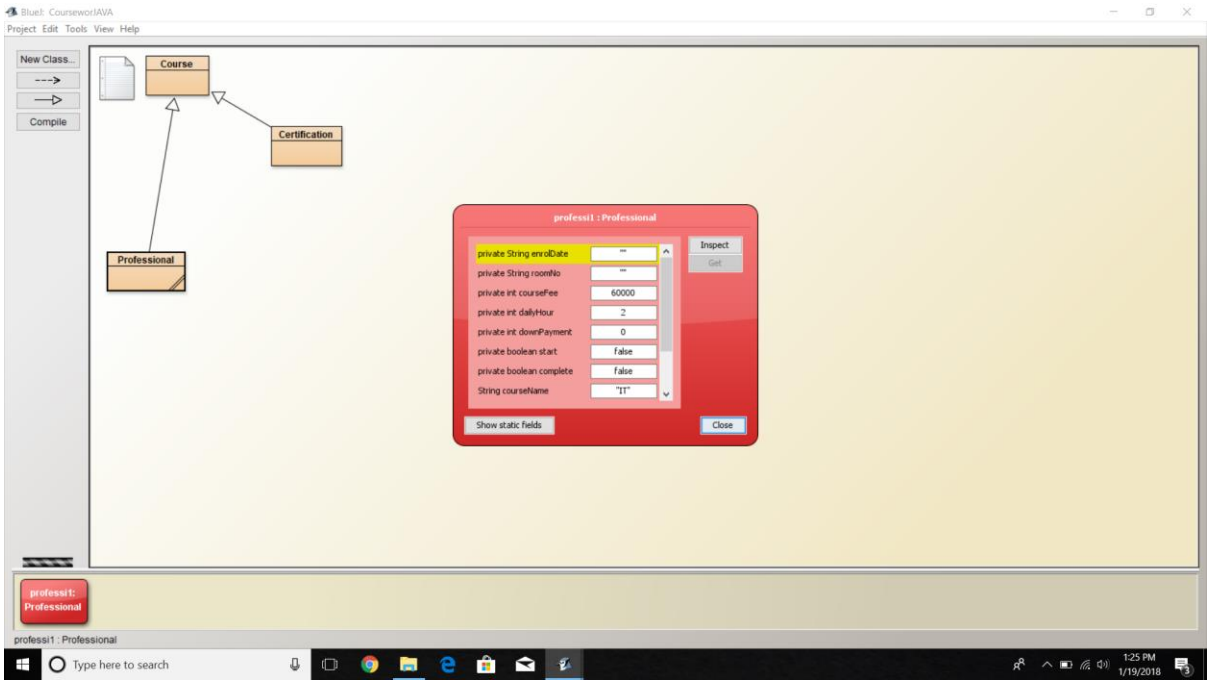
enrolStudent

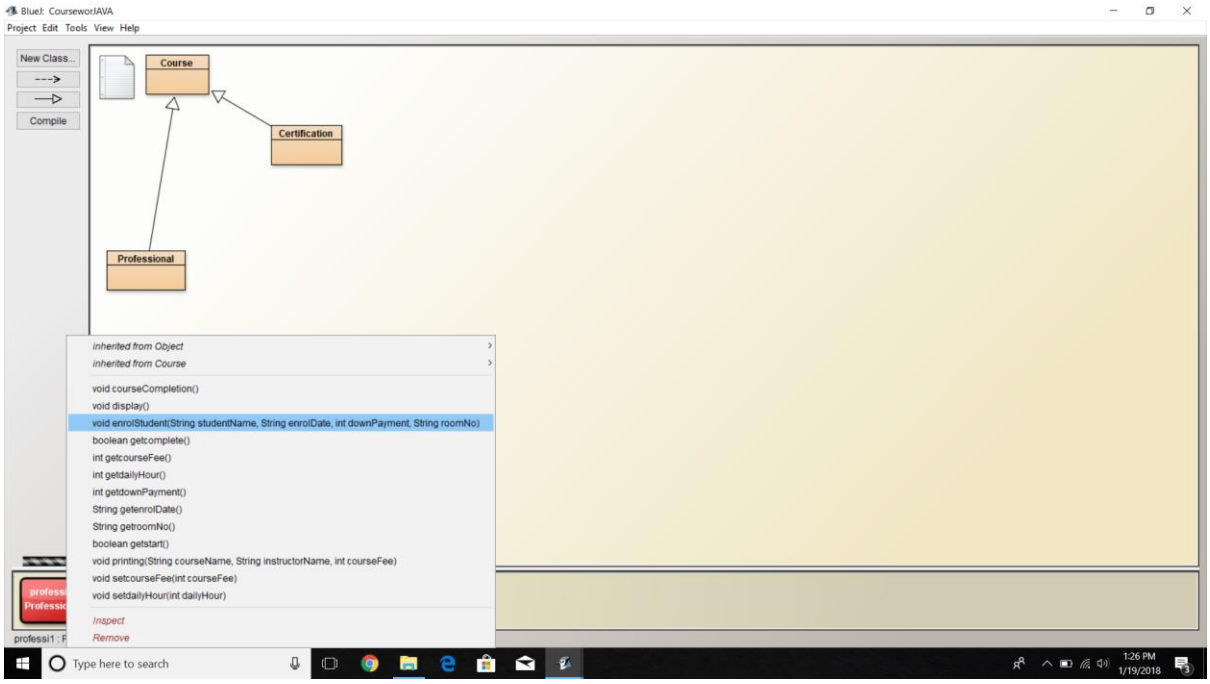
This method is used to enrol new students into the class.

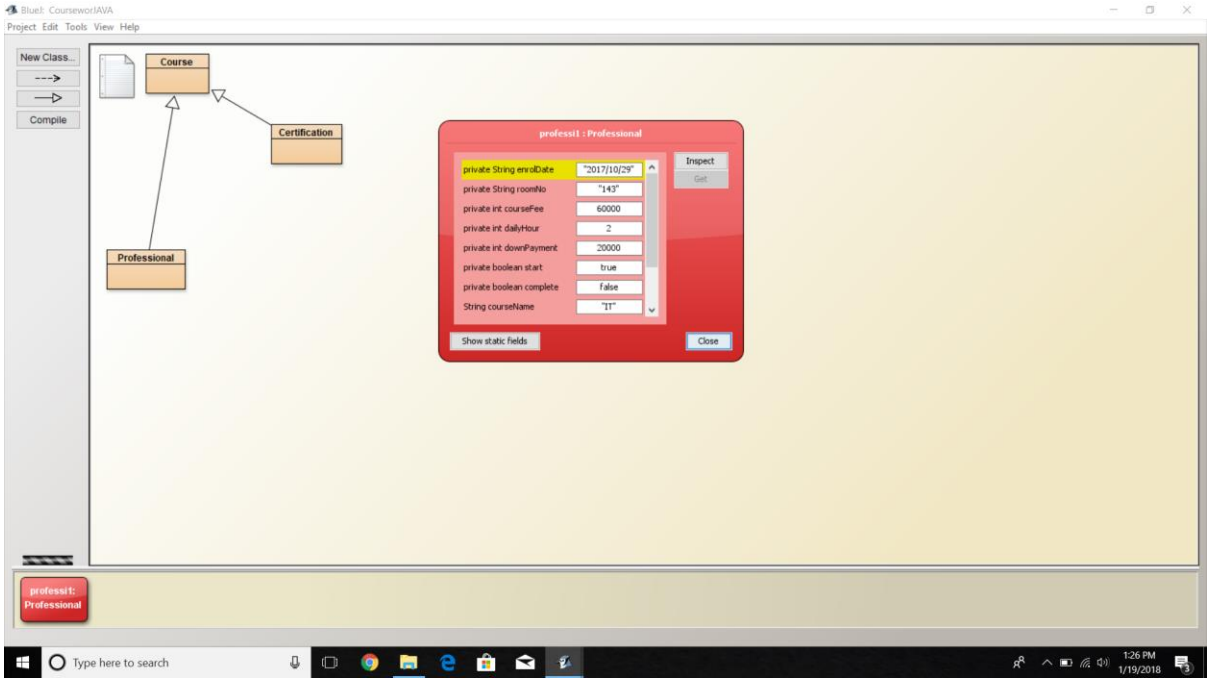
5 Testing

5.1 Test 1

Objective	Creating object, inspecting it, enrolling student and again, reinspecting.
Action	Method is called Following values are added. Object is reinspected.
Expected Result	Object is created, it is inspected, and a student is enrolled and again reinspected.
Actual Result	Object was created, it was inspected, student enrolled and again reinspected.
Conclusion	Test successful.

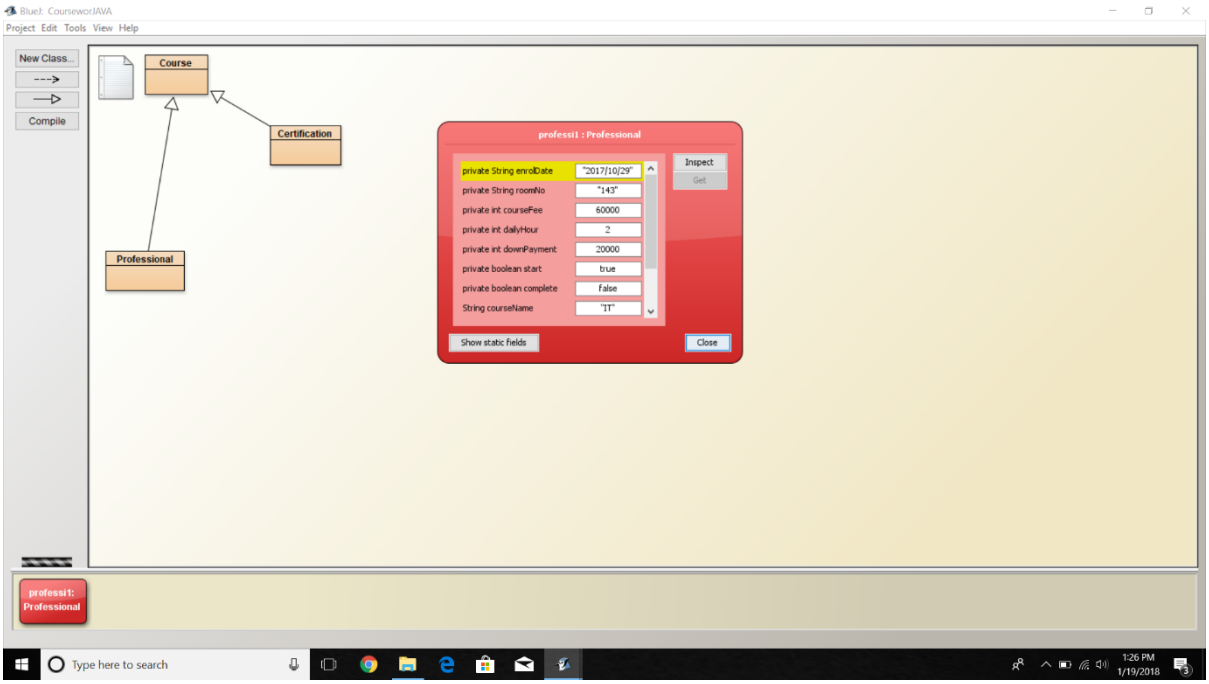


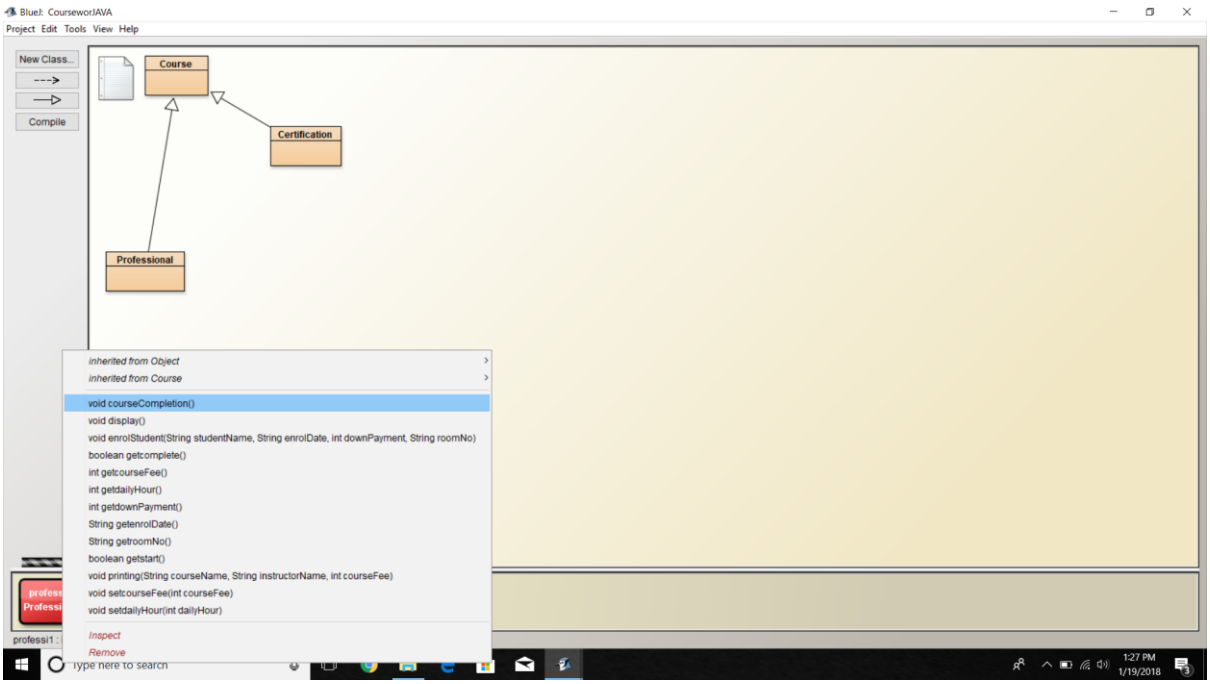

BlueJ: CourseworkJAVA
Project Edit Tools View Help
New Class...
Compile
Course
Certification
Professional
inherited from Object
inherited from Course
void courseCompletion()
void display()
void enrolStudent(String studentName, String enrolDate, int downPayment, String roomNo)
boolean getcomplete()
int getcourseFee()
int getdailyHour()
int getdownPayment()
String getenrolDate()
String getroomNo()
boolean getstart()
void printing(String courseName, String instructorName, int courseFee)
void setcourseFee(int courseFee)
void setdailyHour(int dailyHour)
Inspect
Remove
Type here to search
1:26 PM
1/19/2018

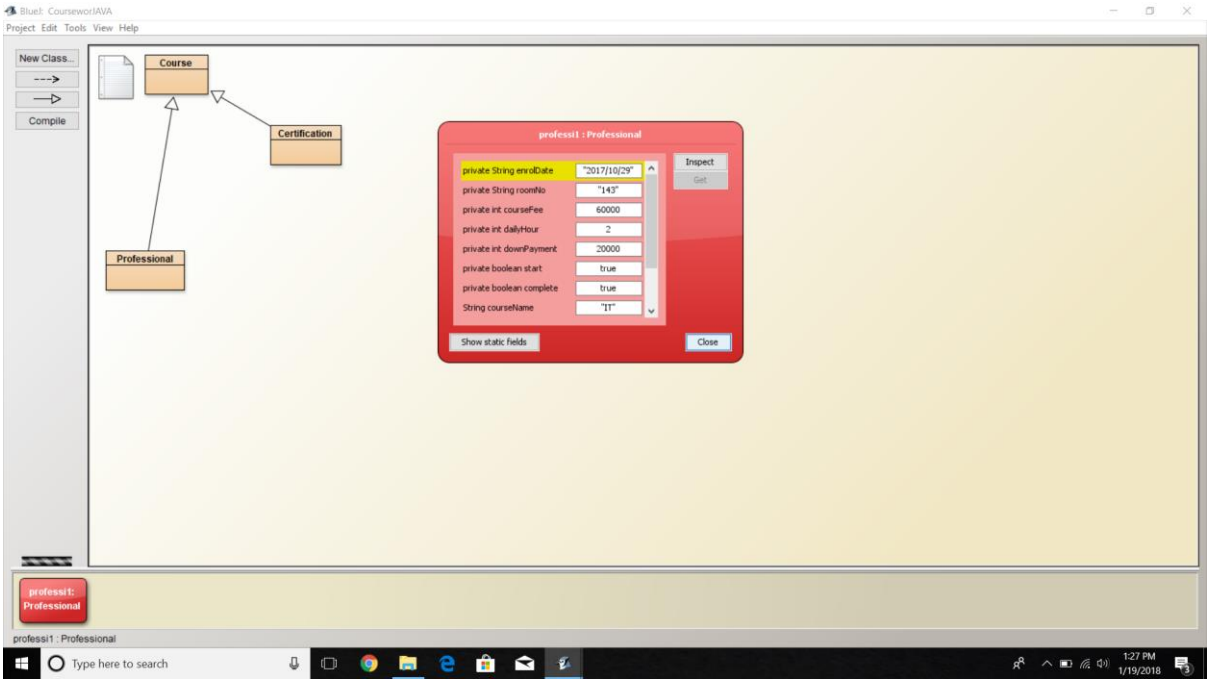

BlueJ: CourseworkJAVA
Project Edit Tools View Help
New Class...
Compile
Course
Certification
Professional
professI : Professional
private String enrolDate "2017/10/29"
private String roomNo "143"
private int courseFee 60000
private int dailyHour 2
private int downPayment 20000
private boolean start true
private boolean complete false
String courseName "IT"
Inspect
Get
Show static fields
Close
professI: Professional
Type here to search
1:26 PM
1/19/2018

5.2 Test 2

Objective	The object created in test 1 is inspected, course is set to complete and the object is reinspected.
Action	Objects are created. Method is called. Course is set to complete.
Expected Result	Course complete is set to true
Actual Result	Course complete was set to true
Conclusion	Test successful.

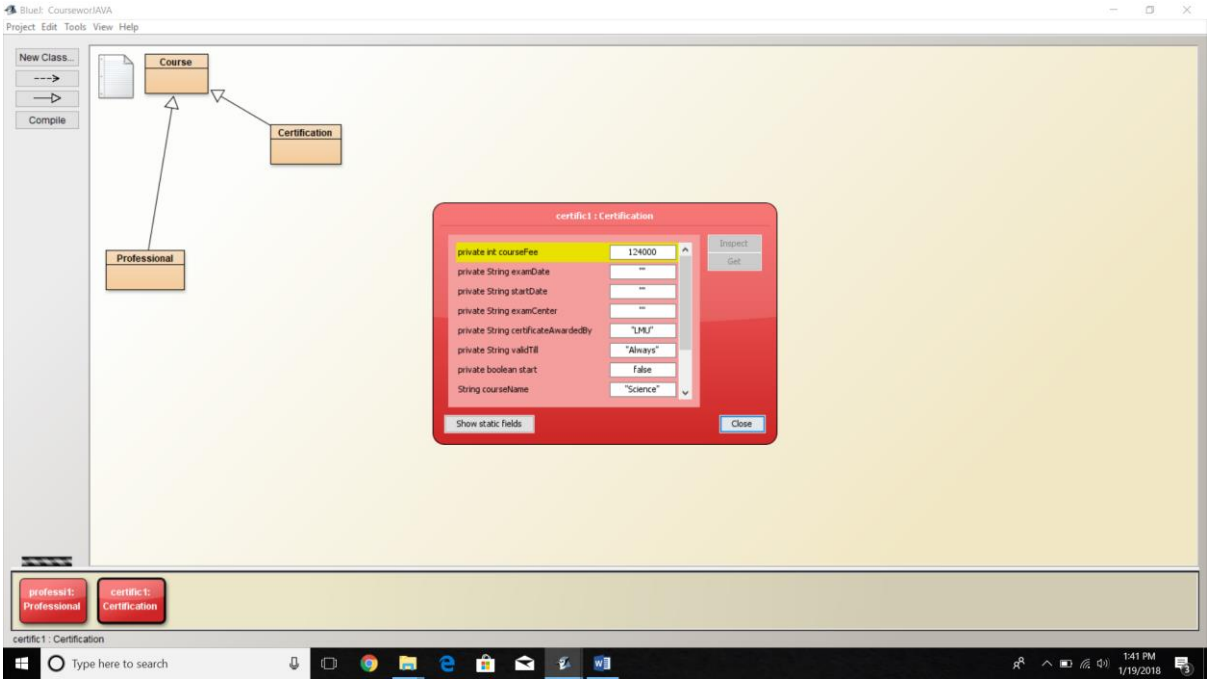

BlueJ: CourseworJAVA
Project Edit Tools View Help
New Class...
Compile
Course
Certification
Professional
professi1 : Professional
private String enrolDate "2017/10/29"
private String roomNo "143"
private int courseFee 60000
private int dailyHour 2
private int downPayment 20000
private boolean start true
private boolean complete false
String courseName "IT"
Inspect
Get
Show static fields
Close
professi1: Professional
Type here to search
1:26 PM 1/19/2018

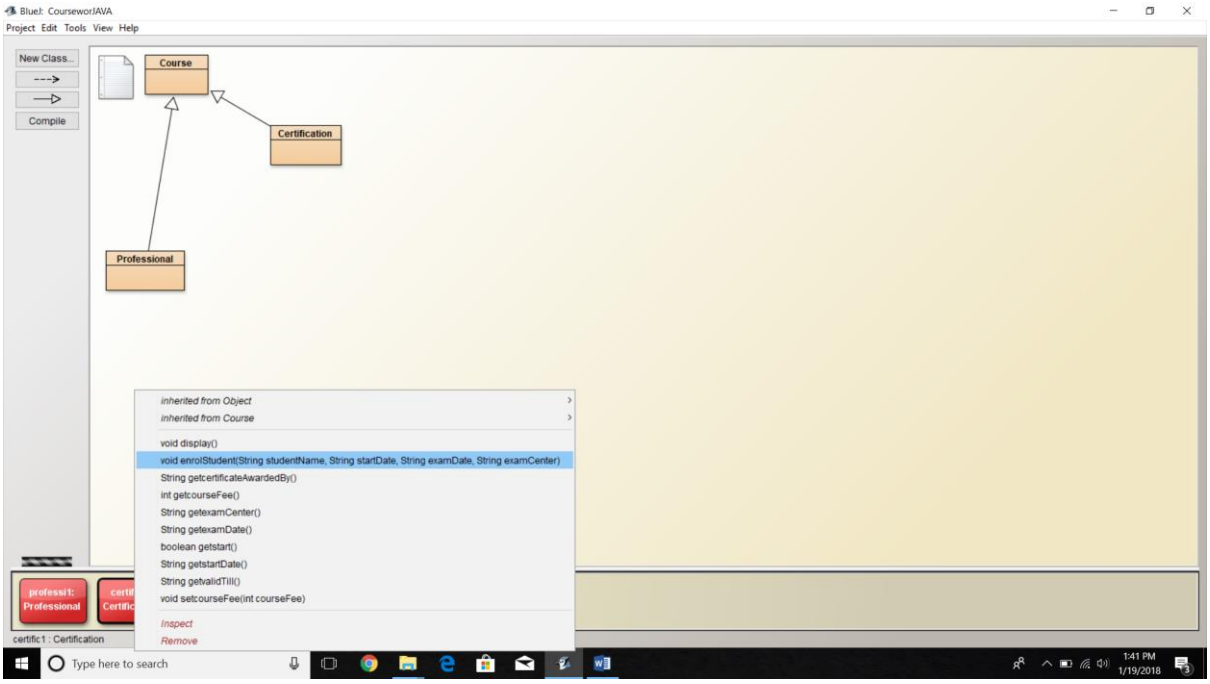

BlueJ: CourseworJAVA
Project Edit Tools View Help
New Class...
Compile
Course
Certification
Professional
inherited from Object
inherited from Course
void courseCompletion()
void display()
void enrolStudent(String studentName, String enrolDate, int downPayment, String roomNo)
boolean getcomplete()
int getcourseFee()
int getdailyHour()
int getdownPayment()
String getenrolDate()
String getroomNo()
boolean getstart()
void printing(String courseName, String instructorName, int courseFee)
void setcourseFee(int courseFee)
void setdailyHour(int dailyHour)
Inspect
Remove
professi1 :
Type here to search
1:27 PM 1/19/2018

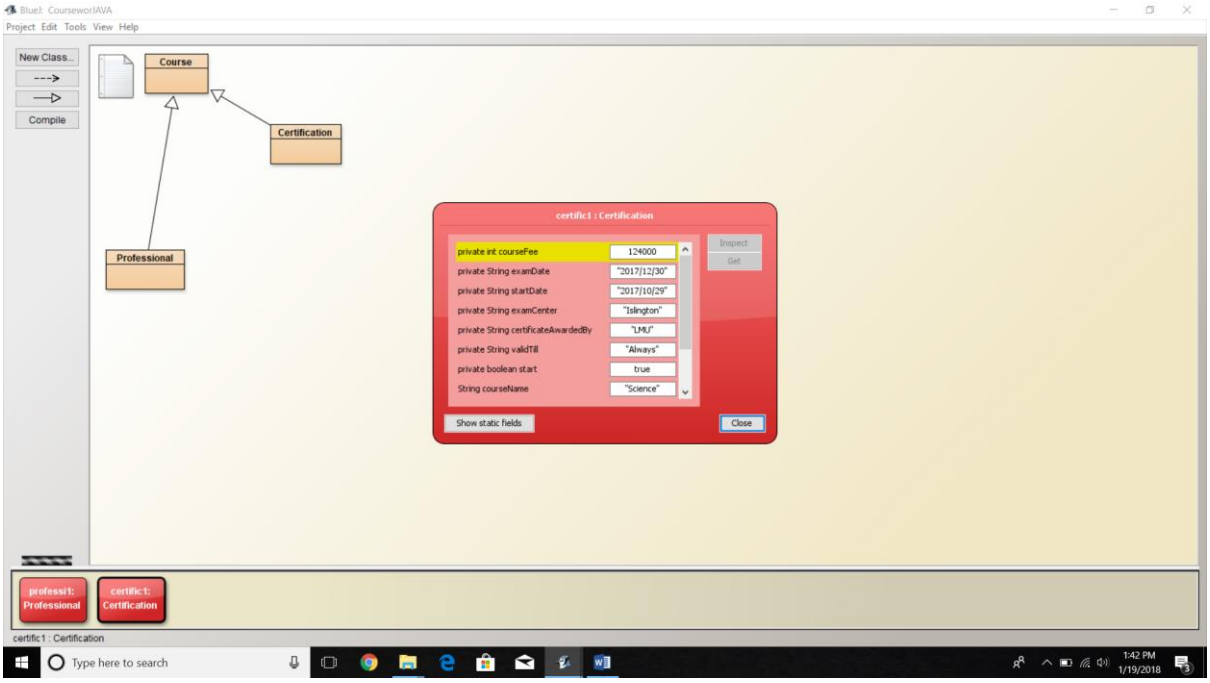

BlueJ: CourseworkAVA
Project Edit Tools View Help
New Class...
Compile
Course
Certification
Professional
professi1 : Professional
private String enrolDate "2017/10/29"
private String roomNo "143"
private int courseFee 60000
private int dailyHour 2
private int downPayment 20000
private boolean start true
private boolean complete true
String courseName "IT"
Inspect
Get
Show static fields
Close
professi1: Professional
professi1 : Professional
Type here to search
1:27 PM
1/19/2018

5.3 Test 3

Objective	Creating object, inspecting, enrolling student and reinspecting.
Action	Object was created. Inspected. Student was enrolled. Reinspected.
Expected Result	Object is created, it is inspected, and a student is enrolled and again reinspected.
Actual Result	Object was created, it was inspected, student enrolled and again reinspected.
Conclusion	Test successful.

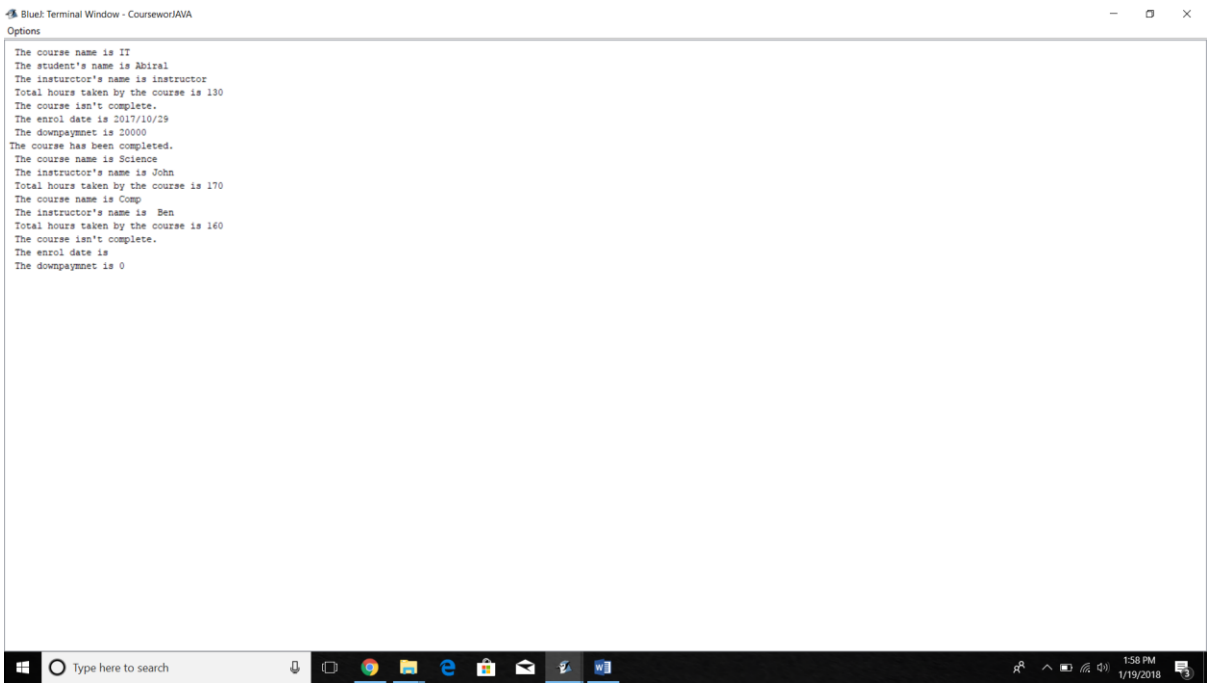



BlueJ: CourseworJAVA
Project Edit Tools View Help
New Class...
Compile
Course
Certification
Professional
inherited from Object
inherited from Course
void display()
void enrolStudent(String studentName, String startDate, String examDate, String examCenter)
String getcertificateAwardedBy()
int getcourseFee()
String getexamCenter()
String getexamDate()
boolean getstart()
String getstartDate()
String getvalidTill()
void setcourseFee(int courseFee)
Inspect
Remove
professi1: Professional
certific1 : Certification
Type here to search
1:41 PM
1/19/2018

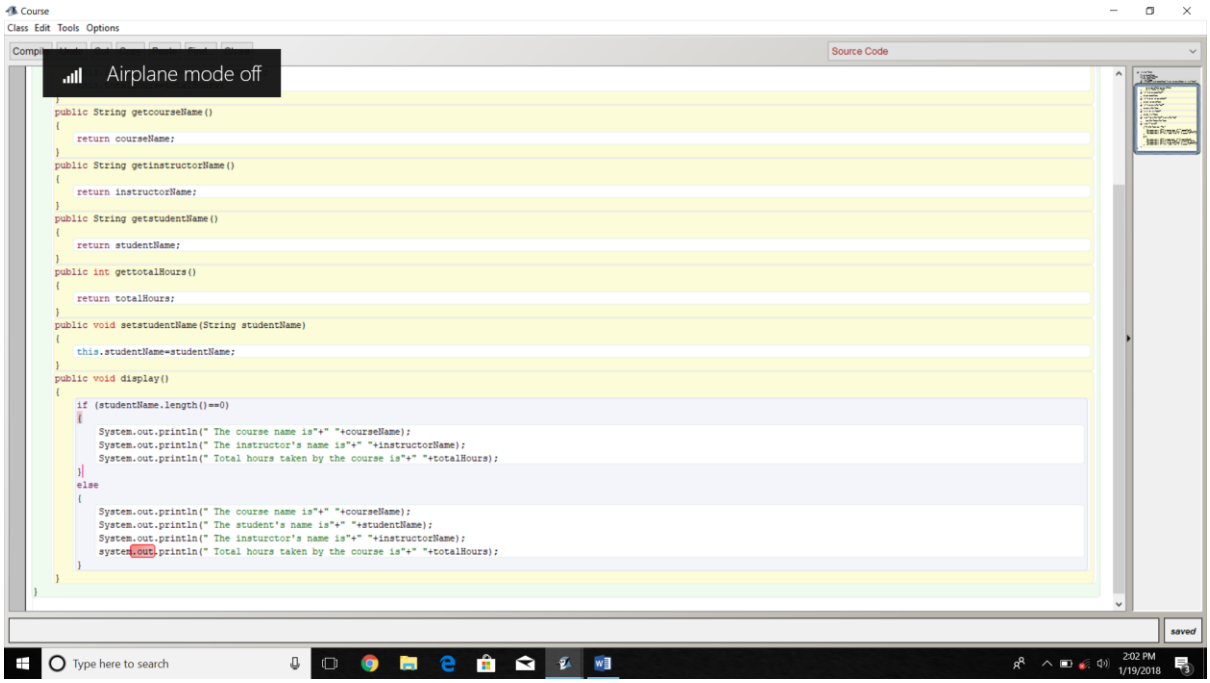

BlueJ: CourseworJAVA
Project Edit Tools View Help
New Class...
Compile
Course
Certification
Professional
certific1 : Certification
private int courseFee 124000
private String examDate "2017/12/30"
private String startDate "2017/10/29"
private String examCenter "Islington"
private String certificateAwardedBy "LMU"
private String validTill "Always"
private boolean start true
String courseName "Science"
Inspect
Get
Show static fields
Close
professi1: Professional
certific1: Certification
certific1 : Certification
Type here to search
1:42 PM
1/19/2018

5.4 Test 4

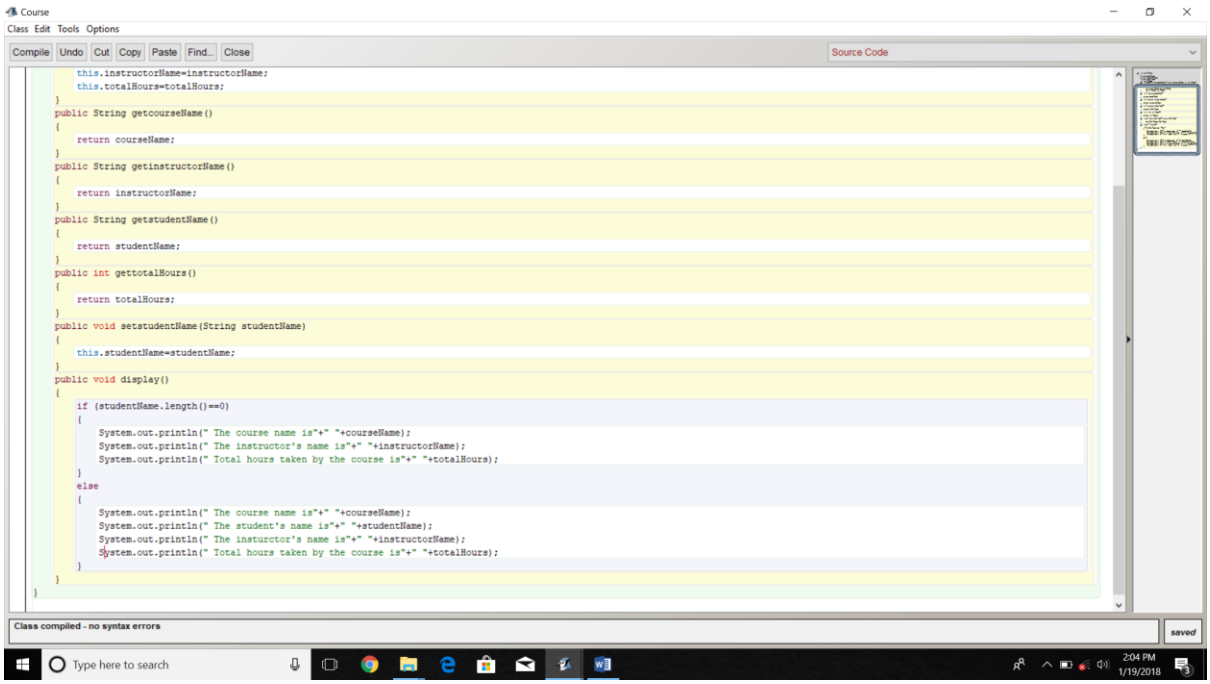
Objective	Create object, inspect it, enrol instructor and reinspect.
Action	Object was created. It was inspected. Instructor was enrolled. It was reinspected.
Expected Result	Object is created, it is inspected, and instructor is enrolled and again reinspected.
Actual Result	Object was created, it was inspected, instructor was enrolled and reinspected.
Conclusion	Test successful.



Errors



Error was found in the syntax of the code. The ‘s’ in System.out.println was an error. And the error was removed.



6 Conclusion

All the tasks in the coursework was finally completed through much trial and errors. The tasks assigned in the coursework were not easy at all. It required lots of labor and research. For the successful completion of all the tasks, each task was carried out in steps and pseudocodes were used. And lastly, the written program was tested to ensure that it had no bugs and errors and delivered accurate result. Finally, after completion of all the assigned tasks, submission was done.

This project didn't only complete all the tasks assigned, but also, helped in developing various skills and taught many things which can be really useful in future career as a programmer. While being involved in this project, sound knowledge of JAVA was obtained. Valuable experience has been gained working on this project.

Although this project was intended for successful completion and submission of all the tasks assigned in the coursework, it doesn't mean it has limited purpose. It can be really useful to all the people who have curiosity about the JAVA programming.