Islington College

Programming CS40001NI

Coursework 1

Submitted By: Rajiv Luitel

ID: NP01CP4A170089

Group: L1 C10

Date: January 18th, 2018

Submitted To:

Mr. Bhim Bahadur Sunar

Lecturer, IT

Contents

Table of figure

Table of table

1. Introduction:

The following coursework is an individual assessment of the project of dealing with the concepts of inheritance where a parent class can have 2 or more than 2 child or sub classes where each child class can access the attributes and methods used in the parent class. The individual assessment was given to us on week 8 and was to completely be completed by the end of week 12 by our module leader Bhim Bahadur Sunar.

The following course work is very much vital in prospect in identifying and gathering information about the concepts of inheritance where a parent class can have 2 or more than 2 child or sub classes where each child class can access the attributes and methods used in the parent class. Furthermore, the course work deals with the concepts of constructors, methods, data types and fields. The task as per the respective course work to be labelled is to create a parent class which has two sub child classes where the child classes can access the methods and attributes. It deals with gathering information, generating information user inputs and displaying them in output about how the course the user was reading and from whom the user learned by spending the total value and being certified using course class as a parent class professional and certification class as a child class using the concepts of inheritances.

2. Class diagram

2.1 course

COURSE
-course Name: String -Instructor Name: String -Student Name: String -Total Hours: Integer
+Course(course Name: String, instructor Name: String, total Hours: Integer) +get_ course Name(): String +get_ instructor Name(): String +get_ student Name(): String +get_ total Hours(): Integer +student Name(new Name: String):void +display(): void

Table 1: Couse class

2.2 Professional

Professional
-course Fee: float -enroll Date: String -room Number: String -daily Hour: float -down Payment: float -started: Boolean -completed: Boolean
+ Professional(course Name: String, instructor Name: String, course Fee: Float, daily Hour: float, total Hours: integer) + get_ enroll Date(): String + get_ room No(): String + get_ down Payment(): float + get_ course Fee(): float + get_ daily Hour(): float + get_ started(): Boolean + get_ completed(): Boolean + new_ course Fee(course Fee: float): void + new_ daily Hour(daily Hour: float): void + enroll Student(student name: String, enroll Date: String, down Payment: float, room Number: String): void + course Completion(): void + print(): void + display(): void

Table 2: Professional class

2.3 certification

Certification
-course Fee: Integer -start Date: String -exam Date: String -exam Centre: String -certificate Awarded By: String -valid Till: String -started: Boolean
+ Certification(course Name: String, instructor Name: String, total Hours: Integer, course Fee: Integer, certificate Awarded By: String, valid Till: String + get_ course Fee(): integer + get_ start Date(): String + get_ exam Date(): String + get_ exam Centre(): String + get_ certificate Awarded by(): String + get_ valid Till(): String + get_ started(): Boolean + set_ course Fee(new Course Fee: Integer): void + enroll Student(student Name: String, start Date: String, exam Date: String, exam Centre: String): void + display(): void

Table 3: Certification class

3. Pseudocode:

3.1 Course class.

```
Create a class named course.
Declare String course Name, String instructor Name, String student Name, integer total Hours attributes.
Declare a constructor having String course Name, String instructor Name, integer total Hours as parameters.
DO
    Assign the values of course Name, Instructor Name, total hours, student name using “this.” Function which represents that we are dealing with the same class where we are currently working on.
END DO
Declare a string return type get method (get_ course Name ())
DO
    Return the value of course name to get_ course Name ().
END DO
Declare a string return type get method (get_ instructor Name ())
DO
    Return the value of instructor name to get_ instructor Name ().
END DO
Declare a string return type get method (get_ student Name ())
DO
    Return the value of student name to get_ student Name ().
END DO
Declare a integer return type get method (get_ total Hours ())
DO
    Return the value of total Hours name to get_ total Hours ().
END DO
Declare a void return type Student name method having String new name as parameter
DO.
```

```
    Set the new name to the student name.
END DO
Declare a display method.
DO
    Print the data stored on course name, total hours and instructor name.
    If (student Name! =" ")
DO
    Print the student name also if the student name is not equals to an empty string.
END DO
End if
END DO
```

3.2 Professional class.

```
Inherit course class in Professional class using extends keyword.
Declare float course Fee, String enroll Date, String room No, float daily Hour, float down Payment, Boolean started, Boolean completed.
Declare a constructor having String course Name, String instructor Name, float course Fee, float daily Hour, integer total Hours as a parameter.
DO
        Call the course Name, instructor Name, total Hours parameters from the super class constructor using super () keyword.
        Assign the values of course fee and daily hours using “this.” Function which represents that we are dealing with the same class where we are currently working on.
        Assign the values of enroll Date to an empty string, assign room No to an empty string,
        Down payment to zero, assign started as false and completed also as a false.
END DO
Declare a string return type get method (get_ enroll Date ())
DO
        Return the value of enroll date to get_ enroll Date ().
END DO
Declare a string return type get method (get_ room No ())
DO
        Return the value of room number to get_ room No ().
END DO
Declare a float return type get method (get_ down Payment ())
DO
        Return the value of down payment to get_ down Payment ().
END DO
Declare a float return type get method (get_ course Fee ())
DO
        Return the value of course fee to get_ course Fee ().
```

```
END DO
Declare a float return type get method (get_ daily Hour ())
DO
    Return the value of daily hour to get_ daily Hour ().
END DO
Declare a Boolean return type get method (get_ started ())
DO
    Return the value of started to get_ started ().
END DO
Declare a Boolean return type get method (get_ completed ())
DO
    Return the value of completed to get_ completed ().
END DO
Declare a void return type new_ course Fee method having float course Fee as parameter.
DO
    Set the value of course fee.
END DO
Declare a void return type new_ daily Hour method having float daily Hour as parameter
DO.
    Set the value of daily hour.
Declare a constructor having String student name, String enroll Date, float down Payment, String room Number as a parameter.
DO
    If started is True.
    DO
        Inform the user by printing the course has already started.
        Print the value of Instructor name
        Print the value of Room number
    END DO
    Else
    DO
        Call the student name constructor from the super class.
        Assign a new enroll date using “.this” keyword.
        Assign a new room number using “.this” keyword.
        Assign a new down payment using “.this” keyword.
        Declare started to True and completed to False.
    End if
    END DO
END DO
END DO
Declare void return type course Completion ()
DO.
    If (completed = True)
    DO
        Print out the course is completed.
    END DO
    Else
    DO
        Call student name from the super class to assign a new name.
        Set the enroll date to an empty string.
        Down payment to zero.
        Started to False.
        Completed to True.
    End if
    END DO
END DO
Declare print method.
DO
    Print the course name form get_ course name method.
    Print the instructor name.
    Print the course fee.
```

```
END DO
Declare a display method.
DO
    Call display method of the super class.
    If started is true.
    DO
        Print the status of completed.
        Print Date of enrollment.
        Print Down payment.
    End if
    END DO
END DO
```

3.3 Certification class

```
Inherit course class in Certification class using extends keyword.
Declare integer course Fee, String start Date, String exam Date, String exam center, String certificate Awarded By, String valid Till, Boolean started.
Declare a constructor having String course Name, String instructor Name, integer total Hours, integer course Fee, String certificate Awarded By, String valid Till as parameters.
DO
    Call the course Name, instructor Name, total Hours parameters from the super class constructor using super () keyword.
    Assign the values of course fee, certification awarded by, valid till using “this.” Function which represents that we are dealing with the same class where we are currently working on.
    Assign start Date to an empty string.
    Assign exam Date to an empty string.
    Assign exam Center to an empty string.
    Assign started as false.
END DO
Declare a string return type get method (get_ course fee ())
DO
    Return the value of course fee to get_ course fee ().
END DO
Declare a string return type get method (get_ start date ())
DO
    Return the value of start date to get_ start Date ().
END DO
Declare a string return type get method (get_ exam Date ())
DO
    Return the value of exam date to get_ exam Date ().
END DO
Declare a string return type get method (get_ exam center ())
DO
```

```
    Return the value of exam center to get_ exam center ().
END DO
Declare a string return type get method (get_ certification awarded by ())
DO
    Return the value of certification awarded by to get_ certification awarded by ().
END DO
Declare a string return type get method (get_ valid till ())
DO
    Return the value of valid till to get_ valid till ().
END DO
Declare a Boolean return type get method (get_ started ())
DO
    Return the value of started to get_ started ().
END DO
Declare a void return type set_ course Fee method having integer new course Fee as
parameter.
DO
    If not started
    DO
        Set the value of Course Fee.
    END DO
    Else
    DO
        Print "The certification course is already started therefore course fee is
        Impossible to change".
    END DO
    End if
END DO
Declare a void return type enroll Student method having String student Name, String
start Date, String exam Date, String exam Center parameter.
DO
```

```
    If not started
    DO
        Call student Name method having student Name as a parameter.
        Set started to true.
    END DO
    Else
    DO
        Print The course was already started on.
    END DO
        Assign the values of exam date and exam center using “this.” Function
        which represents that we are dealing with the same class where we are
        currently working on.
    End if
END DO
Declare a display method.
DO
    Call the display method from the super class.
    If started is true
    DO
        Print the Start date.
        Print the Student's name
        Print the Exam date.
        Print the Exam center.
        Print the Certificate awarded by.
        Print the Certification validity duration.
    End if
    END DO
END DO
```

4. Description of each methods.

4.1 For course (parent)

Public class course:

This method is used to create a class named course.

Public Course (String course Name, String instructor Name, integer total Hours):

This method is used to create a constructor having 3 parameters.

Public String get_ course Name ():

This method is used to obtain or store the values returned by course name.

Public String get_ instructor Name ():

This method is used to obtain or store the values returned by instructor name.

Public String get_ student Name ():

This method is used to obtain or store the values returned by student name.

Public integer get_ total Hours ():

This method is used to obtain or store the values returned by total hours.

Public void student Name (String new Name):

This method is used to set a new student’s name.

Public void display ():

This method displays the texts written under the display method.

4.2 For professional (child)

Public class Professional extends Course:

This method inherits the attributes and methods of the course class so that it can be used on the Professional class.

Public Professional (String course Name, String instructor Name, float course Fee, float daily Hour, integer total Hours):

This method is used to create a constructor having 5 parameters.

Public String get_ enroll Date ():

This method is used to obtain or store the values returned by enroll date.

Public String get_ room Number ():

This method is used to obtain or store the values returned by room number.

Public float get_ down Payment ():

This method is used to obtain or store the values returned by down payment.

Public float get_ course Fee ():

This method is used to obtain or store the values returned by course fee.

Public float get_ daily Hour ():

This method is used to obtain or store the values returned by daily hour.

Public Boolean get_ started ():

This method is used to obtain or store the values returned by started.

Public Boolean get_ completed ():

This method is used to obtain or store the values returned by completed.

Public void new_ course Fee (float course Fee):

This method is used to set a new course fee.

Public void new_ daily Hour (float daily Hour):

This method is used to set a new daily hour.

Public void enroll Student (String student name, String enroll Date, float down Payment, String room Number).

This method is used to create a method named enroll student having 4 parameters.

Public void course Completion ():

This method is used to create a method named course completion so that it can proceed further in 2 different patterns that is if the course is completed and next one is if it is not completed.

Public void print ():

This method is used to print out the text under the print method.

Public void display ():

This method is used to display out the text under the display method.

4.3 For Certification (child)

Public class certification extends Course:

This method inherits the attributes and methods of the course class so that it can be used on the certification class.

Public Certification (String course Name, String instructor Name, integer total Hours, integer course Fee, String certificate Awarded By, String valid Till):

This method is used to create a constructor having 6 parameters.

Public integer get_ course Fee ():

This method is used to obtain or store the values returned by course fee.

Public String get_ start Date ():

This method is used to obtain or store the values returned by start date.

Public String get_ exam Date ():

This method is used to obtain or store the values returned by exam date.

Public String get_ exam Centre ():

This method is used to obtain or store the values returned by exam center.

Public String get_ certificate Awarded by ():

This method is used to obtain or store the values returned by certificate Awarded by.

Public String get_ valid Till ():

This method is used to obtain or store the values returned by valid till.

Public Boolean get_ started ():

This method is used to obtain or store the values returned by started.

Public void set_ course Fee (integer new Course Fee):

This method is used to create a method named set_ course Fee so that it can proceed further in 2 different patterns that is if the course not started set the new course fee and if the course has started display a message.

Public void enroll Student (String student Name, String start Date, String exam Date, String exam Centre):

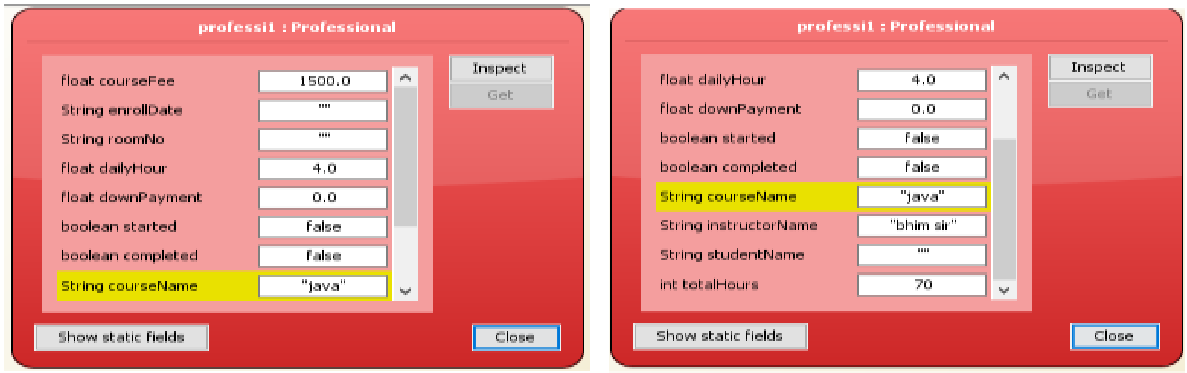
This method is used to create a method named enroll student having 4 parameters.

Public void display ():

This method is used to display out the text under the display method.

5. Test:

5.1 Test 1:



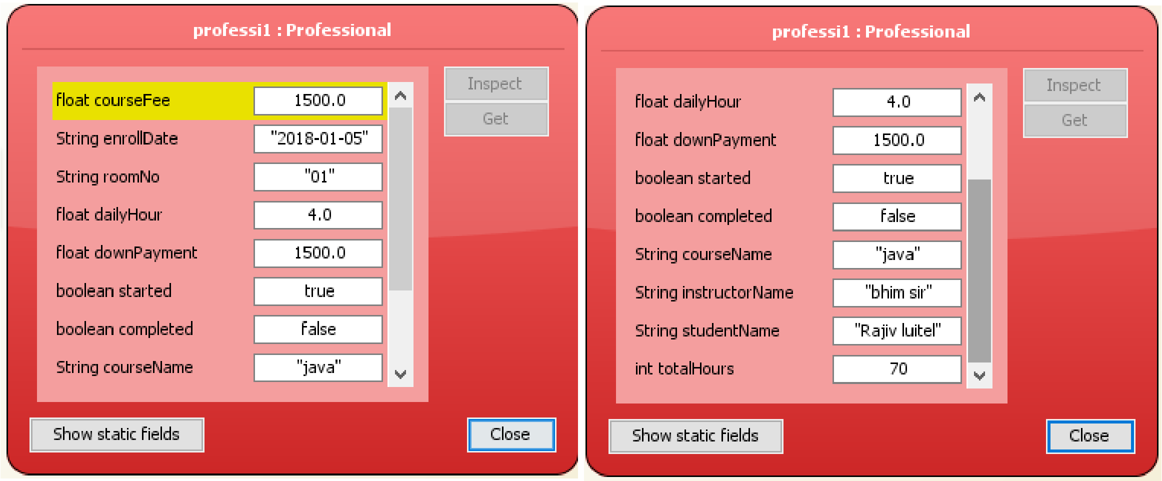
professi1 : Professional

```
float courseFee            1500.0
String enrollDate          ""
String roomNo              ""
float dailyHour            4.0
float downPayment          0.0
boolean started            false
boolean completed          false
String courseName          "java"
```

professi1 : Professional

```
float dailyHour            4.0
float downPayment          0.0
boolean started            false
boolean completed          false
String courseName          "java"
String instructorName      "bhim sir"
String studentName         ""
int totalHours             70
```

Figure 1 Inspecting Professional class



professi1 : Professional

```
float courseFee            1500.0
String enrollDate          "2018-01-05"
String roomNo              "01"
float dailyHour            4.0
float downPayment          1500.0
boolean started            true
boolean completed          false
String courseName          "java"
```

professi1 : Professional

```
float dailyHour            4.0
float downPayment          1500.0
boolean started            true
boolean completed          false
String courseName          "java"
String instructorName      "bhim sir"
String studentName         "Rajiv luitel"
int totalHours             70
```

Figure 2 Inspecting Professional class after enrolling student for the course

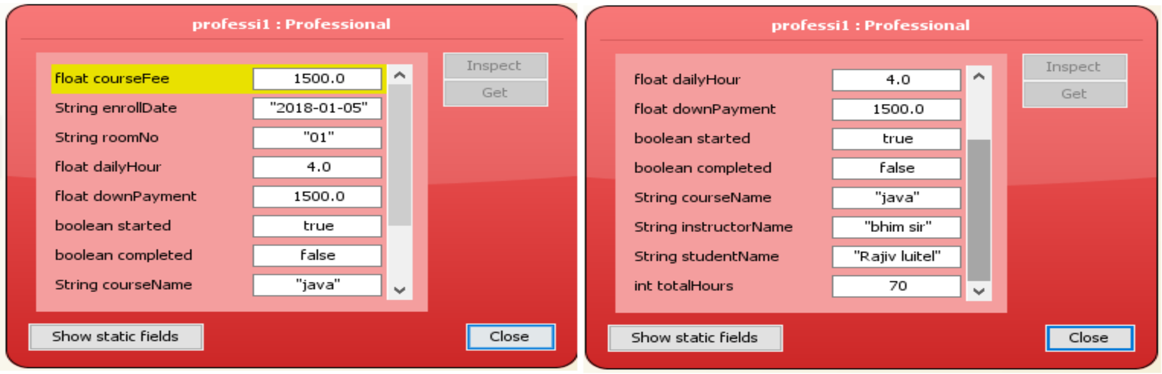
Objective	To inspect the professional class, enrol student for that particular course and re-inspect professional class.

Action	Constructor is called and values are assigned to the parameters.
	Professional class is inspected.
	enrolStudent method is called;
	studentName= “Rajiv luitel”
	enrolDate= “2018-01-05”
	roomNo= “01”
	downPayment= 1500
	Professional class is re-inspected

Expected Result	Object will be created and the details of the student will be assigned.
Actual Result	Object is created and the details of the student are assigned.
Conclusion	Test successful.

Table 4: Enroll student testing

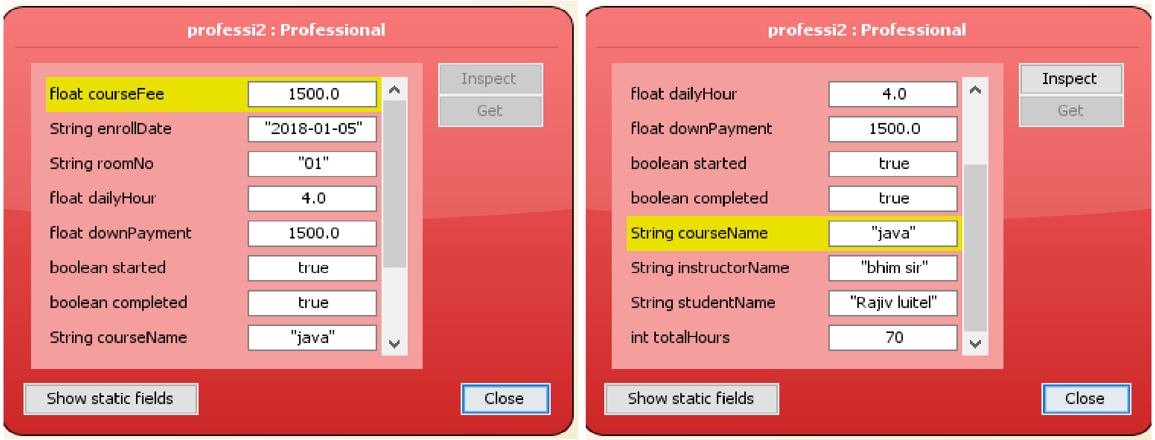
5.2 Test 2:



professi1 : Professional

float courseFee	1500.0
String enrollDate	"2018-01-05"
String roomNo	"01"
float dailyHour	4.0
float downPayment	1500.0
boolean started	true
boolean completed	false
String courseName	"java"
String instructorName	"bhim sir"
String studentName	"Rajiv luitel"
int totalHours	70

Figure 3 Inspecting Professional class



professi2 : Professional

float courseFee	1500.0
String enrollDate	"2018-01-05"
String roomNo	"01"
float dailyHour	4.0
float downPayment	1500.0
boolean started	true
boolean completed	true
String courseName	"java"
String instructorName	"bhim sir"
String studentName	"Rajiv luitel"
int totalHours	70

Figure 4 Inspecting Professional class after changing course to completed.

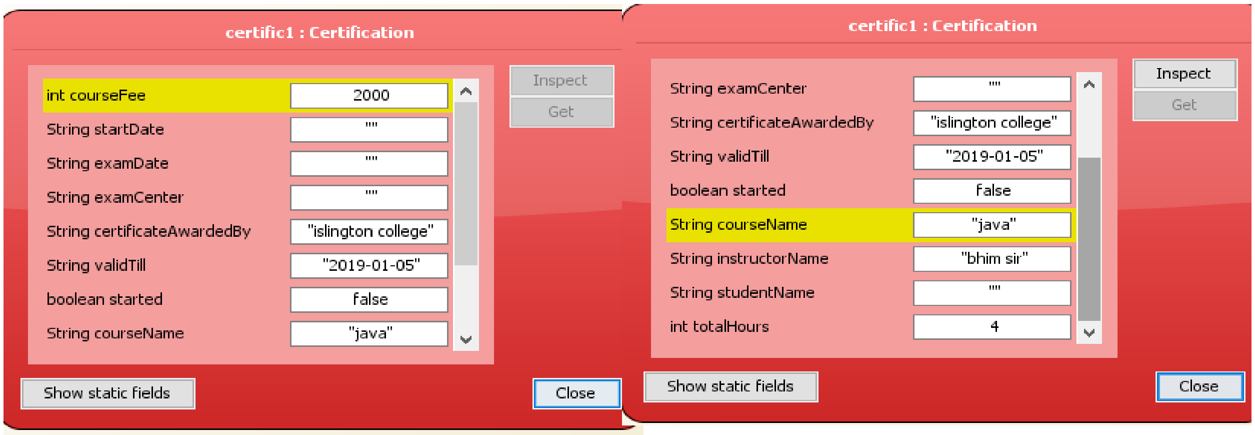
Object	To inspect Professional class, change the status of course to complete and re-inspect Professional class.

Action	1. Constructor is called and values are assigned to the parameters.
	2. enrolStudent method is called;
	studentName= “Rajiv luitel”
	enrolDate= “2018-01-05”
	roomNo= “01”
	downPayment= 1500
	3. courseCompletion method is called.
	4. Professional class is re-inspected.

Expected Result	The status of the course completion will be changed to true.
Actual Result	The status of the course completion is change to true.
Conclusion	Test successful

Table 5: Course complete test

5.3 Test 3:



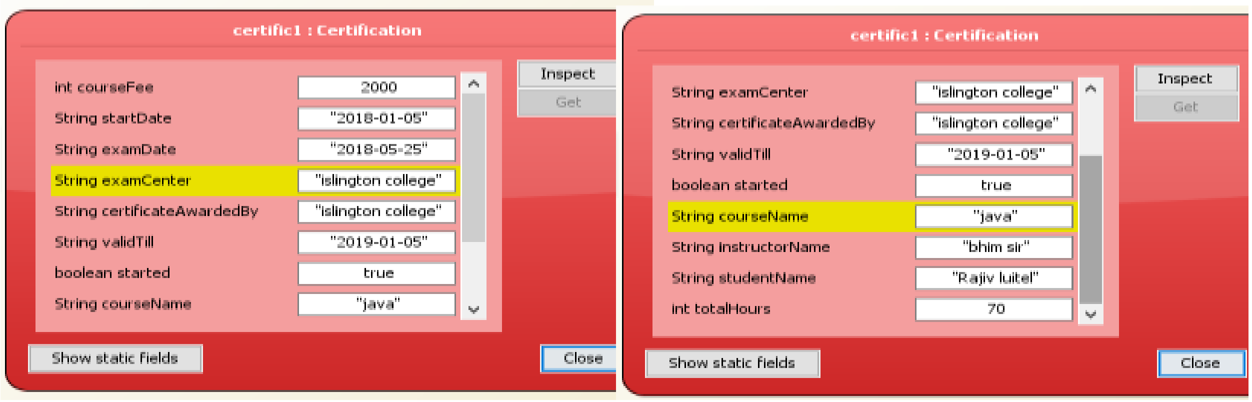
certific1 : Certification

int courseFee 2000
String startDate ""
String examDate ""
String examCenter ""
String certificateAwardedBy "islington college"
String validTill "2019-01-05"
boolean started false
String courseName "java"

certific1 : Certification

String examCenter ""
String certificateAwardedBy "islington college"
String validTill "2019-01-05"
boolean started false
String courseName "java"
String instructorName "bhim sir"
String studentName ""
int totalHours 4

Figure 5 Inspecting Certification class



certific1 : Certification

int courseFee 2000
String startDate "2018-01-05"
String examDate "2018-05-25"
String examCenter "islington college"
String certificateAwardedBy "islington college"
String validTill "2019-01-05"
boolean started true
String courseName "java"

certific1 : Certification

String examCenter "islington college"
String certificateAwardedBy "islington college"
String validTill "2019-01-05"
boolean started true
String courseName "java"
String instructorName "bhim sir"
String studentName "Rajiv luitel"
int totalHours 70

Figure 6 Inspecting Certification class after enrolling student.

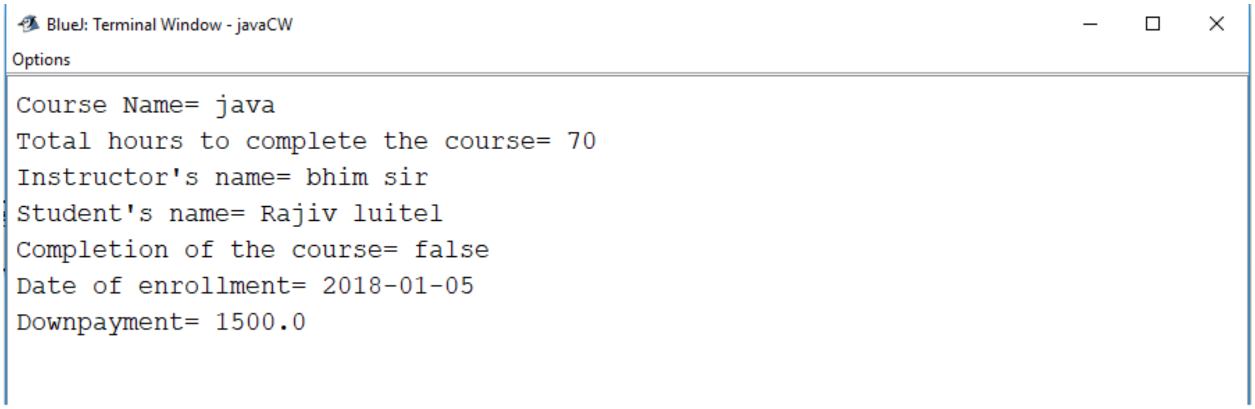
Objective	To inspect Certification class, enrol a student and re-inspect Certification class

Action	1. Constructor is called of certification class and values are assigned to the parameters.
	2. enrolStudent method is called;
	studentName= “Rajiv luitel”
	startDate= “2018-01-05”
	examDate= “2018-05-25”
	examCenter= “Islington College”

Expected Result	Detail of student and the course exam will be assigned.
Actual Result	Detail of student and the course exam is assigned.
Conclusion	Test successful.

Table 6: certification class enroll student test

5.4 Test 4:



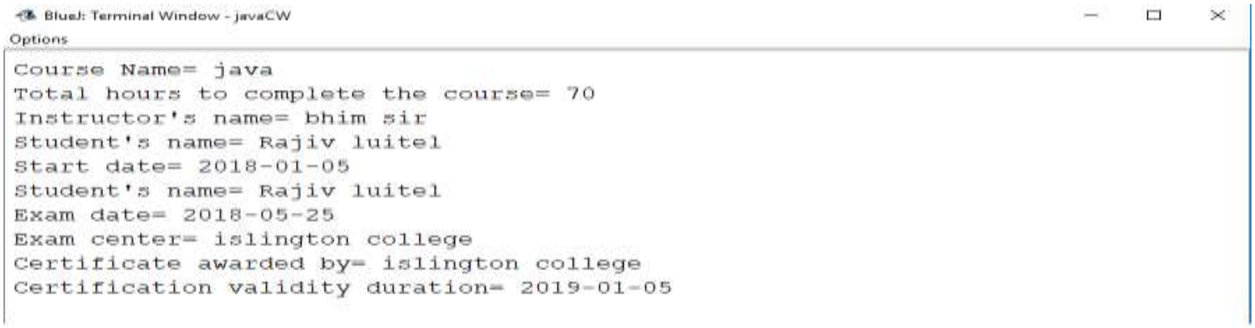
```
Course Name= java
Total hours to complete the course= 70
Instructor's name= bhim sir
Student's name= Rajiv luitel
Completion of the course= false
Date of enrollment= 2018-01-05
Downpayment= 1500.0
```

Figure 7: Display detail of professional class

Objective	To display the detail of Professional class and Certification class
Action	1.Constructors are called of both the classes and values are assigned to the parameters. 2. enrolStudent method is called and the instance variables are assigned the value of corresponding parameters of both classes. 3. display method is called of both the classes.
Expected Result	The details of the Professional class and Certification class will be displayed.
Actual Result	The details of the Professional class and Certification class are displayed.
Conclusion	Test successful.

Table 7: display method test

5.5 Test 5:



```
Course Name= java
Total hours to complete the course= 70
Instructor's name= bhim sir
Student's name= Rajiv luitel
Start date= 2018-01-05
Student's name= Rajiv luitel
Exam date= 2018-05-25
Exam center= islington college
Certificate awarded by= islington college
Certification validity duration= 2019-01-05
```

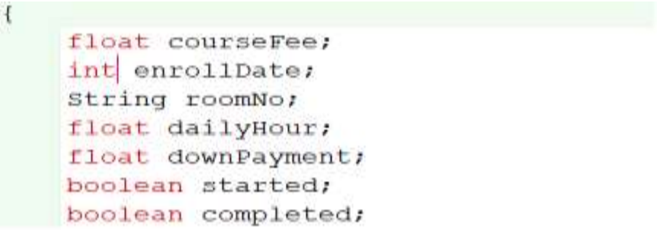
Figure 8: Display detail of certification class

Objective	To display the detail of Professional and Certification Class
Action	1. Constructor is called of both the child classes and values are assigned to the parameters. 2. enrollStudents method is called from both the classes and value assigned accordingly.
Expected Result	The details of the course and the students will be assigned in the Professional class. The detail of the certificate will be assigned in the Certification class.
Actual Result	The detail of the course and the students is assigned in the Professional class. The detail of the certificate is assigned in the Certification class.
Conclusion	Test successful.

Table 8: Display method test.

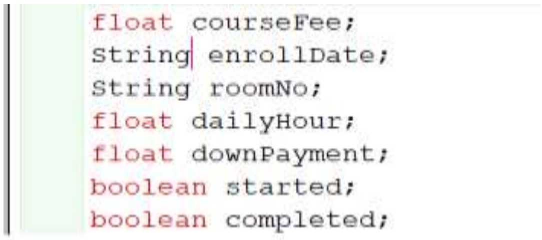
6. Error detection:

6.1Error1



```
{
    float courseFee;
    int enrollDate;
    String roomNo;
    float dailyHour;
    float downPayment;
    boolean started;
    boolean completed;
```

Figure 9: enrollDate declared as integer data type



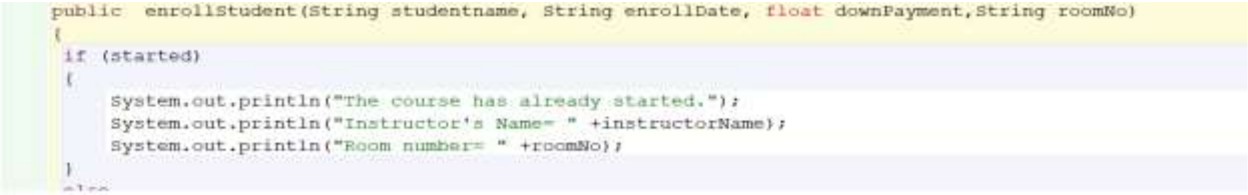
```
    float courseFee;
    String enrollDate;
    String roomNo;
    float dailyHour;
    float downPayment;
    boolean started;
    boolean completed;
```

Figure 10: Data type of enrollDate changed to string

Error	Integer number too large.
Cause	The value of enrolment date of a student is too large for an integer value. Integer data type cannot store number that is large as a date.
Correction	The data type of the enrolment date is changed to string. The string value can hold any value of any length as it holds integer value as a character of a string.

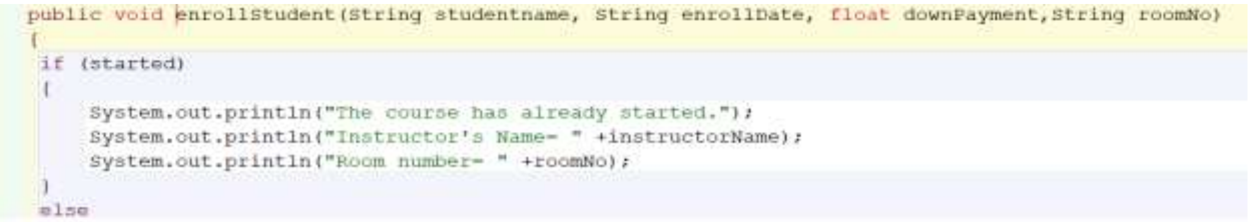
Table 9:Error 1

6.2 Error2



```
public  enrollStudent(String studentname, String enrollDate, float downPayment,String roomNo)
{
 if (started)
 {
     System.out.println("The course has already started.");
     System.out.println("Instructor's Name= " +instructorName);
     System.out.println("Room number= " +roomNo);
 }
 else
```

Figure 11: No void return type declared improper use of the method.



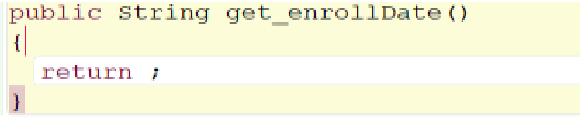
```
public void enrollStudent(String studentname, String enrollDate, float downPayment,String roomNo)
{
 if (started)
 {
     System.out.println("The course has already started.");
     System.out.println("Instructor's Name= " +instructorName);
     System.out.println("Room number= " +roomNo);
 }
 else
```

Figure 12: void return type declared proper use of the method.

Error	Improper declaration of the method.
Cause	The method which is designed to be accessed only on that particular class should have a non-return type method that is void.
Correction	The method should have a non-return type void to make it accessible to that particular class only.

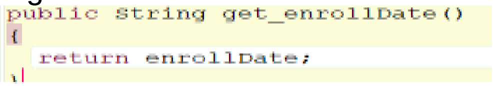
Table 10: Error2

6.3 Error3



```
public String get_enrollDate()
{
  return ;
}
```

Figure 13: Return method used without attribute to return the value



```
public String get_enrollDate()
{
  return enrollDate;
}
```

Figure 14: Attribute is added so that the return statement can return the value.

Error	Improper declaration of return statement.
Cause	Return statement is used to return the value of the attribute to the get method but without attribute on the return statement the get method has no values to hold.
Correction	The return method should have attributes so that the get method can hold or have a value to store.

Table 11: Error3

7. Conclusion

The following coursework of java programming is to provide us deep concepts of the use of inheritance and proper use of the constructors, data structures and methods used in java programming language.it also helps to make a program suitable environment in which a student takes on a course and from whom he studied the course and from whom or what he was certified. It has been Enlighted practice guiding the way to understand how java programming works on java and what kind of tools are to be used according to the environment.

Class having subclasses has been taught on our course and so were the data types and data structures which has made it easier, since the topic was already familiar it was easy to deal with and research more on it.

There was a hard time when many obstacles were on my way I had no idea how to conduct on the coursework, about how to write, how to submit, how to complete it. After many try finally I was able to obtain an outcome. The course was a very difficult journey so far. It was a serious challenge but necessary to go through coursework. This coursework helped me practically to know about the data types and about the classes and sub classes. If I were to solve the same course without having much time and preparation, and at my previous base knowledge I would not have been able to complete the course within the enlisted time period.

Sub classes are inherited to classes and the inputs the given information to class. All the information that is given to the subclasses are accessible to the main class program. The individual assessment was not just for the course but for knowledge and practice with java programming. From this course I have required proper knowledge about main classes and sub classes, inheritance, public value, private value etc. I was also acknowledged about making test file, pseudocode, writing method description etc. Putting aside all the hard work and labour done to complete the coursework it was very challenging journey where I got the great opportunities to tackle with the java based problem and one step ahead in the world of java programming.