

This **assignment** uses three classes which we use in programs we *make, manipulate and access* objects. These objects were specified in the lab text, and model a university setting, involving courses, students and lecturers. Below are the classes I used in my programs.

## (1) Degree Class

```
/**
 *
 * Lab 13, Degree Class
 *
 * Written by Gareth Evans
 *
 * 11/03/99
 */
public class Degree
{
    private String DegreeName;

    public void setDegreeName(String tempname)
    {
        DegreeName = tempname;
    }

    public String getDegreeName()
    {
        return DegreeName;
    }
}
```

## (2) Lecturer Class

```
/**
 *
 * Lab 13, Lecturer Class
 *
 * Written by Gareth Evans
 *
 * 11/03/99
 */

public class Lecturer
{

    private String LecturerName;
    private String ResearchArea;
    private Degree CourseOrganizer;

    public void setLecturerName(String tempname)
    {
        LecturerName = tempname;
    }

    public String getLecturerName()
    {
        return LecturerName;
    }

    public void setResearchArea(String temparea)
    {
        ResearchArea = temparea;
    }

    public String getResearchArea()
    {
        return ResearchArea;
    }

    public void setCourseOrganizer(Degree tempcourse)
    {
        CourseOrganizer = tempcourse;
    }

    public Degree getCourseOrganizer()
    {
        return CourseOrganizer;
    }
}
```

### (3) Student Class

```
/**
 *
 * Lab 13, Student Class
 *
 * Written by Gareth Evans
 *
 * 11/03/99
 */

public class Student
{

    private String Name;
    private int StudyYear;
    private Degree Course;
    private Lecturer Supervisor;

    public void setName(String tempname)
    {
        Name = tempname;
    }

    public String getName()
    {
        return Name;
    }

    public void setStudyYear(int tempyear)
    {
        StudyYear = tempyear;
    }

    public int getStudyYear()
    {
        return StudyYear;
    }

    public void setCourse(Degree tempcourse)
    {
        Course = tempcourse;
    }

    public Degree getCourse()
    {
        return Course;
    }

    public void setSupervisor(Lecturer tempsupervisor)
    {
        Supervisor = tempsupervisor;
    }

    public Lecturer getSupervisor()
    {
        return Supervisor;
    }

}
```

**Part A:** Create an **ARRAY** of 2 degrees (\*NOT\* 2 individual objects!). The degrees are CSE and CSB (standing for *Computer Systems Engineering* and *Computer Systems with Business Studies* respectively).

## Program Code

```
/**
 *
 * Lab 13, Part A
 *
 * Written by Gareth Evans
 *
 * 11/03/99
 */

import java.bangor.*;

public class PartA
{
    /**
     *
     * main
     *
     */

    public static void main(String arg[])
    {
        // Define the Variables

        Degree[] DegreeList;
        DegreeList = new Degree[2];
        for (int temp1 = 0; temp1 < 2; temp1++)
        {
            DegreeList[temp1] = new Degree();
        }

        // Set the Degree Values

        DegreeList[0].setDegreeName("CSE");
        DegreeList[1].setDegreeName("CSB");

        // end of code
    }
}
```

**Part B:** Create an **ARRAY** of 3 lecturers (\*NOT\* 3 individual objects!). A. Brown, *Software Engineering*; C. Dillon, *Object-Oriented Design*; E. France, *Database Systems*. A. Brown is the **organizer** of CSE. C. Dillon and and E. France are the **co-organizers** of CSB

## Program Code

```
/**
 *
 * Lab 13, Part B
 *
 * Written by Gareth Evans
 *
 * 11/03/99
 */

import java.bangor.*;

public class PartB
{
    /**
     *
     * main
     *
     */

    public static void main(String arg[])
    {
        // Define the Variables

        // Define the Degree List

        Degree[] DegreeList;
        DegreeList = new Degree[2];
        for (int temp1 = 0; temp1 < 2; temp1++)
        {
            DegreeList[temp1] = new Degree();
        }

        // Define the Lecturer List

        Lecturer[] LecturerList;
        LecturerList = new Lecturer[3];
        for (int temp1 = 0; temp1 < 3; temp1++)
        {
            LecturerList[temp1] = new Lecturer();
        }
    }
}
```

```

// Set the Degree Values

DegreeList[0].setDegreeName("CSE");
DegreeList[1].setDegreeName("CSB");

// Set the Lecturer Values

LecturerList[0].setLecturerName("A. Brown");
LecturerList[0].setResearchArea("Software Engineering");
LecturerList[0].setCourseOrganizer(DegreeList[0]);

LecturerList[1].setLecturerName("C. Dillon");
LecturerList[1].setResearchArea("Object Orientated Design");
LecturerList[1].setCourseOrganizer(DegreeList[1]);

LecturerList[2].setLecturerName("E. France");
LecturerList[2].setResearchArea("Database Systems");
LecturerList[2].setCourseOrganizer(DegreeList[1]);

// end of code
}
}

```

**Part C:** Create an **ARRAY** of 6 students (*\*NOT\** 6 individual objects!). The *first 4 students* study CSB, and the *other 2* study CSE. **A. Brown** supervises students 1, 2. **C. Dillon** supervises students 3, 4. **E. France** supervises students 5, 6. The *rest* of the student information can be set to **any** desired value.

## Program Code

```

/**
 *
 * Lab 13, Part C
 *
 * Written by Gareth Evans
 *
 * 11/03/99
 *
 */

import java.bangor.*;

public class PartC
{
    /**
     *
     * main
     *
     */

    public static void main(String arg[]) throws Exception
    {

```

```

// Define the Degree List

Degree[] DegreeList = new Degree[2];

for (int temp1 = 0; temp1 < 2; temp1++)
    {
        DegreeList[temp1] = new Degree();
    }

// Define the Lecturer List

Lecturer[] LecturerList = new Lecturer[3];
for (int temp1 = 0; temp1 < 3; temp1++)
    {
        LecturerList[temp1] = new Lecturer();
    }

// Define the Student List

Student[] StudentList = new Student[6];
for (int temp1 = 0; temp1 < 6; temp1++)
    {
        StudentList[temp1] = new Student();
    }

// Set the Degree Values

DegreeList[0].setDegreeName("CSE");
DegreeList[1].setDegreeName("CSB");

// Set the Lecturer Values

LecturerList[0].setLecturerName("A. Brown");
LecturerList[0].setResearchArea("Software Engineering");
LecturerList[0].setCourseOrganizer(DegreeList[0]);

LecturerList[1].setLecturerName("C. Dillon");
LecturerList[1].setResearchArea("Object Orientated Design");
LecturerList[1].setCourseOrganizer(DegreeList[1]);

LecturerList[2].setLecturerName("E. France");
LecturerList[2].setResearchArea("Database Systems");
LecturerList[2].setCourseOrganizer(DegreeList[1]);

// Set the Student Values

StudentList[0].setName("Gareth Evans");
StudentList[0].setStudyYear(0);
StudentList[0].setCourse(DegreeList[1]);
StudentList[0].setSupervisor(LecturerList[0]);

StudentList[1].setName("William Jones");
StudentList[1].setStudyYear(0);
StudentList[1].setCourse(DegreeList[1]);
StudentList[1].setSupervisor(LecturerList[0]);

StudentList[2].setName("Mathew Williams");
StudentList[2].setStudyYear(1);
StudentList[2].setCourse(DegreeList[1]);
StudentList[2].setSupervisor(LecturerList[1]);

```

```

StudentList[3].setName("Aaron White");
StudentList[3].setStudyYear(0);
StudentList[3].setCourse(DegreeList[1]);
StudentList[3].setSupervisor(LecturerList[1]);

StudentList[4].setName("Arwel Stevens");
StudentList[4].setStudyYear(2);
StudentList[4].setCourse(DegreeList[0]);
StudentList[4].setSupervisor(LecturerList[2]);

StudentList[5].setName("Thomas Griffiths");
StudentList[5].setStudyYear(3);
StudentList[5].setCourse(DegreeList[0]);
StudentList[5].setSupervisor(LecturerList[2]);

// end of code
}
}

```

**Part D:** Create a **main menu** for your program which lists the following *options*:

1. *Student supervision*
2. *Degree students*
3. *Supervisors*
4. *Supervisors' responsibilities*

What each of your choices should display on the screen is described in the **lab text**.

## Program Code

```

/**
 *
 * Lab 13, Part D
 *
 * Written by Gareth Evans
 *
 * 11/03/99 - 18/03/99
 *
 */

import java.bangor.*;

public class PartD
{

// Define Global Variables

static Lecturer[] LecturerList;
static Degree[] DegreeList;
static Student[] StudentList;

```

```
// Define Methods for use in the Main Method...
```

```
public static void ShowStudentSupervision()
{
    System.out.println();

    for (int showtemp = 0; showtemp < 6; showtemp++)
    {
        System.out.print(StudentList[showtemp].getName());

        // Show the name of the student pointed to by the Student List
        // element

        System.out.print(" is supervised by ");
        System.out.println(StudentList[showtemp].getSupervisor()
                            .getLecturerName());

        /*
         *   Get the Supervisor of the student.
         *
         *   Note: StudentList[showtemp].getSupervisor() returns a
         *           Supervisor object, which we can then perform on
         *           the .getLecturerName method to obtain the name
         *           of the Supervisor/Lecturer.
         */
    }
}
```

```
public static void ShowDegreeStudents()
{
    System.out.println();

    for (int degreeshow = 0; degreeshow < 2; degreeshow++)
    {
        System.out.print("The following students attend ");
        System.out.println(DegreeList[degreeshow].getDegreeName());

        // The above line prints out a Degree Name in the Degree array

        for (int test = 0; test < 6; test++)
        {
            if (StudentList[test].getCourse().getDegreeName() ==
                DegreeList[degreeshow].getDegreeName())
            {
                System.out.println(StudentList[test].getName());
            }

            /*
             *   If, in the student list, a student's degree is the same
             *   as the degree we printed out above, then we print
             *   the name of the student
             */
        }

        System.out.println();
    }
}
```

```

public static void ShowSupervisors()
{
    System.out.println();

    for (int lecturershow = 0; lecturershow < 3; lecturershow++)
    {
        System.out.print(LecturerList[lecturershow].getLecturerName());

        // Print out a lecturer name in the list

        System.out.println(" Supervises the following students:");

        for (int test = 0; test < 6; test++)
        {
            if (StudentList[test].getSupervisor().getLecturerName() ==
                LecturerList[lecturershow].getLecturerName())

                /*
                 * If A Student's lecturer in the Student list
                 * Matches the Lecturer we printed out earlier,
                 * then do the following code (Print out the Student's name
                 */

                {
                    System.out.println(StudentList[test].getName());
                }

        }

        System.out.println();
    }
}

```

```

public static void ShowResponsibilities() throws Exception
{
    // Define Local Variables

    int not_a_student = 0;
    String ChoiceName = new String();

    // Code...

    System.out.println();

    System.out.println("Please enter the name of a student.");
    ChoiceName = BasicIo.readString();

    for (int check = 0; check < 6; check++)
    {
        if (StudentList[check].getName().equals(ChoiceName))

            // If a student's name in the list matches the input String:

            {
                System.out.print(ChoiceName);
                System.out.print("'s supervisor organises the ");
                System.out.print(StudentList[check].getSupervisor()
                    .getCourseOrganizer().getDegreeName());
                System.out.println(" Course");

                not_a_student = 1;
            }
    }
}

```

```

        if (not_a_student < 1)
        {
            System.out.println(ChoiceName + " is not a student");
        }
    }

    /**
     *
     * main
     *
     */

public static void main(String arg[]) throws Exception
{

    // Define the Variables

    int Choice;

    // Define the Degree List

    DegreeList = new Degree[2];

    for (int temp1 = 0; temp1 < 2; temp1++)
    {
        DegreeList[temp1] = new Degree();
    }

    // Define the Lecturer List

    LecturerList = new Lecturer[3];
    for (int temp1 = 0; temp1 < 3; temp1++)
    {
        LecturerList[temp1] = new Lecturer();
    }

    // Define the Student List

    StudentList = new Student[6];
    for (int temp1 = 0; temp1 < 6; temp1++)
    {
        StudentList[temp1] = new Student();
    }

    // Set the Degree Values

    DegreeList[0].setDegreeName("CSE");
    DegreeList[1].setDegreeName("CSB");

    // Set the Lecturer Values

    LecturerList[0].setLecturerName("A. Brown");
    LecturerList[0].setResearchArea("Software Engineering");
    LecturerList[0].setCourseOrganizer(DegreeList[0]);

    LecturerList[1].setLecturerName("C. Dillon");
    LecturerList[1].setResearchArea("Object Orientated Design");
    LecturerList[1].setCourseOrganizer(DegreeList[1]);

    LecturerList[2].setLecturerName("E. France");
    LecturerList[2].setResearchArea("Database Systems");
    LecturerList[2].setCourseOrganizer(DegreeList[1]);

```

```

// Set the Student Values
StudentList[0].setName("Gareth Evans");
StudentList[0].setStudyYear(0);
StudentList[0].setCourse(DegreeList[1]);
StudentList[0].setSupervisor(LecturerList[0]);

StudentList[1].setName("William Jones");
StudentList[1].setStudyYear(0);
StudentList[1].setCourse(DegreeList[1]);
StudentList[1].setSupervisor(LecturerList[0]);

StudentList[2].setName("Mathew Williams");
StudentList[2].setStudyYear(1);
StudentList[2].setCourse(DegreeList[1]);
StudentList[2].setSupervisor(LecturerList[1]);

StudentList[3].setName("Aaron White");
StudentList[3].setStudyYear(0);
StudentList[3].setCourse(DegreeList[1]);
StudentList[3].setSupervisor(LecturerList[1]);

StudentList[4].setName("Arwel Stevens");
StudentList[4].setStudyYear(2);
StudentList[4].setCourse(DegreeList[0]);
StudentList[4].setSupervisor(LecturerList[2]);

StudentList[5].setName("Thomas Griffiths");
StudentList[5].setStudyYear(3);
StudentList[5].setCourse(DegreeList[0]);
StudentList[5].setSupervisor(LecturerList[2]);

// MENU PART OF PROGRAM
do // repeat this code until the user wants to exit
{
    System.out.println();
    System.out.println("MAIN MENU - PLEASE ENTER YOUR CHOICE");
    System.out.println("-----");
    System.out.println();
    System.out.println("1. Student Supervision");
    System.out.println("2. Degree Students");
    System.out.println("3: Supervisors");
    System.out.println("4. Supervisors' Responsibilities");
    System.out.println();
    System.out.println("5. Exit");
    System.out.println();
    Choice = BasicIo.readInteger();

    switch(Choice)
    {
        case 1: ShowStudentSupervision();
                break;

        case 2: ShowDegreeStudents();
                break;

        case 3: ShowSupervisors();
                break;

        case 4: ShowResponsibilities();
                break;

    }
}
while ((Choice<5) && (Choice>0));
// end of code
}
}

```

# Program Print Out

M:\cafe\Projects Semester 2\Lab13, Part D>java PartD  
Symantec Java! JustInTime Compiler Version 210.050 for JDK 1.1  
Copyright (C) 1996-97 Symantec Corporation

MAIN MENU - PLEASE ENTER YOUR CHOICE  
-----

1. Student Supervision
2. Degree Students
- 3: Supervisors
4. Supervisors' Responsibilities
  
5. Exit

1

Gareth Evans is supervised by A. Brown  
William Jones is supervised by A. Brown  
Mathew Williams is supervised by C. Dillon  
Aaron White is supervised by C. Dillon  
Arwel Stevens is supervised by E. France  
Thomas Griffiths is supervised by E. France

MAIN MENU - PLEASE ENTER YOUR CHOICE  
-----

1. Student Supervision
2. Degree Students
- 3: Supervisors
4. Supervisors' Responsibilities
  
5. Exit

2

The following students attend CSE  
Arwel Stevens  
Thomas Griffiths

The following students attend CSB  
Gareth Evans  
William Jones  
Mathew Williams  
Aaron White

MAIN MENU - PLEASE ENTER YOUR CHOICE  
-----

1. Student Supervision
2. Degree Students
- 3: Supervisors
4. Supervisors' Responsibilities
  
5. Exit

3

A. Brown Supervises the following students:

Gareth Evans  
William Jones

C. Dillon Supervises the following students:

Mathew Williams  
Aaron White

E. France Supervises the following students:

Arwel Stevens  
Thomas Griffiths

MAIN MENU - PLEASE ENTER YOUR CHOICE

-----

1. Student Supervision
2. Degree Students
- 3: Supervisors
4. Supervisors' Responsibilities
  
5. Exit

4

Please enter the name of a student.

Gareth Evans  
Gareth Evans's supervisor organises the CSE Course

MAIN MENU - PLEASE ENTER YOUR CHOICE

-----

1. Student Supervision
2. Degree Students
- 3: Supervisors
4. Supervisors' Responsibilities
  
5. Exit

4

Please enter the name of a student.

Gareth Williams  
Gareth Williams is not a student

MAIN MENU - PLEASE ENTER YOUR CHOICE

-----

1. Student Supervision
2. Degree Students
- 3: Supervisors
4. Supervisors' Responsibilities
  
5. Exit

5

**Part E:** Now create a new **METHOD** in the Degree class called `studentNumber()` which returns the number of students for each degree. Add the following *item* to your menu: **5. Degree student numbers**, which, when selected, returns the following information:

5.

--

Computer Systems with Business Studies: 4 students

Computer Systems Engineering: 2 students

## Degree Class for Part E

```
/**
 *
 * Lab 13, Degree Class
 * Written by Gareth Evans
 * 11/03/99
 *
 * Modified for Part E on 18/03/99
 *
 */
public class Degree
{
    private String DegreeName;
    public void setDegreeName(String tempname)
    {
        DegreeName = tempname;
    }
    public String getDegreeName()
    {
        return DegreeName;
    }
    public int studentNumber(Student[] TempStudentList)
    // We import as a parameter the Student List
    {
        int numberofstudents = 0;
        for (int i = 0; i < 6; i++)
        {
            if (DegreeName ==
                TempStudentList[i].getCourse().getDegreeName())
            /*
             *
             * IF the Degree Name of this Degree object (i.e. the one
             * who called this method) matches a degree a student is taking
             * then we increment the number of students taking the Degree
             *
             */
            {
                numberofstudents++;
            }
        }
        return numberofstudents;
    }
}
```

The program code for Part E is **identical** to the code for part D, apart from two changes. The *first* is that an extra method has been inserted before the main method. The second is that the *menu system* has been changed. The following is a cut-down version of the source code, where “...” indicates code that is **identical** to the code for Part C.

## Program Code

```
/**
 *
 * Lab 13, Part E
 *
 * Written by Gareth Evans
 *
 * 11/03/99 - 18/03/99
 *
 * Like Part D but with altered Degree class
 * and Extra Menu Option
 *
 */

import java.bangor.*;

public class PartE
{
    // Define Global Variables

    static Lecturer[] LecturerList;
    static Degree[] DegreeList;
    static Student[] StudentList;

    // Define Methods for use in the Main Method...

    public static void ShowStudentSupervision()
    {
        ...
    }

    public static void ShowDegreeStudents()
    {
        ...
    }

    public static void ShowSupervisors()
    {
        ...
    }

    public static void ShowResponsibilities() throws Exception
    {
        ...
    }
}
```

```

public static void ShowNumbers()
{
    System.out.println();

    for (int numbershow = 0; numbershow < 2; numbershow++)
    {
        System.out.print(DegreeList[numbershow].getDegreeName());
        System.out.print(": ");

        // The above line prints out a Degree Name in the Degree array

        System.out.print(DegreeList[numbershow].studentNumber(StudentList));

        // The above line uses the studentNumber method in the degree
        // class to get the number of students taking the degree.

        System.out.println(" Student(s)");
    }
}

```

```

/**
 *
 * main
 *
 */

```

```

public static void main(String arg[]) throws Exception
{
    // Define the Variables
    ...

    // Define the Degree List
    ...

    // Define the Lecturer List
    ...

    // Define the Student List
    ...

    // Set the Degree Values
    ...

    // Set the Lecturer Values
    ...

    // Set the Student Values
    ...
}

```

```
// MENU PART OF PROGRAM
```

```
do // repeat this code until the user wants to exit
```

```
{
```

```
    System.out.println();
```

```
    System.out.println("MAIN MENU - PLEASE ENTER YOUR CHOICE");
```

```
    System.out.println("-----");
```

```
    System.out.println();
```

```
    System.out.println("1. Student Supervision");
```

```
    System.out.println("2. Degree Students");
```

```
    System.out.println("3: Supervisors");
```

```
    System.out.println("4. Supervisors' Responsibilities");
```

```
    System.out.println("5: Degree Student Numbers");
```

```
    System.out.println();
```

```
    System.out.println("6. Exit");
```

```
    System.out.println();
```

```
    Choice = BasicIo.readInteger();
```

```
    switch(Choice)
```

```
    {
```

```
        case 1: ShowStudentSupervision();  
                break;
```

```
        case 2: ShowDegreeStudents();  
                break;
```

```
        case 3: ShowSupervisors();  
                break;
```

```
        case 4: ShowResponsibilities();  
                break;
```

```
        case 5: ShowNumbers();  
                break;
```

```
    }
```

```
    }
```

```
while ((Choice<6) && (Choice>0));
```

```
// end of code
```

```
}
```

```
}
```

# Program Print Out

M:\cafe\Projects Semester 2\Lab13, Part E>java PartE  
Symantec Java! JustInTime Compiler Version 210.050 for JDK 1.1  
Copyright (C) 1996-97 Symantec Corporation

MAIN MENU - PLEASE ENTER YOUR CHOICE  
-----

1. Student Supervision
2. Degree Students
- 3: Supervisors
4. Supervisors' Responsibilities
- 5: Degree Student Numbers

6. Exit

2

The following students attend CSE  
Arwel Stevens  
Thomas Griffiths

The following students attend CSB  
Gareth Evans  
William Jones  
Mathew Williams  
Aaron White

MAIN MENU - PLEASE ENTER YOUR CHOICE  
-----

1. Student Supervision
2. Degree Students
- 3: Supervisors
4. Supervisors' Responsibilities
- 5: Degree Student Numbers

6. Exit

5

CSE: 2 Student(s)  
CSB: 4 Student(s)

MAIN MENU - PLEASE ENTER YOUR CHOICE  
-----

1. Student Supervision
2. Degree Students
- 3: Supervisors
4. Supervisors' Responsibilities
- 5: Degree Student Numbers

6. Exit

6