```
/*
 * This is an solution for the  program which forms the second
 * Practical Skills Assessment on COM102.
 * Created by: Niall Morrissey (B00787301), Michael Brown (B00808857)
 * Date: 18/04/2021
 * Version: 1.0
 */

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

/**
 * This class holds the main functionality of the program.
 * It initialises the program, reads in & processes both student and course
 * detail text files, and provides a menu system for the user to navigate.
 */

public class Main
   {
   private int selection;
   private Scanner scan = new Scanner(System.in);

   public static void main(String[] args)
      {
      //Instantiating self class to declare instance variables and
      //avoid the need to make more methods static.
      Main enrolmentRegister = new Main();
      enrolmentRegister.initialise();
      }
```

```
  private void initialise()
    {
    //Instantiating Student_Register class to populate Student array
    //and call Student_Register methods within navigation menu
    Student_Register studentObject = new Student_Register();

    // open the text file
    Scanner studentTxtFile = null;
    Scanner courseTxtFile = null;
    try
      {
      studentTxtFile = new Scanner(new File("StudentDetails.txt"));
      courseTxtFile = new Scanner(new File("CourseDetails.txt"));
      }
    catch (FileNotFoundException ex)
      {
      System.out.println(ex.getMessage());
      System.out.println("in " + System.getProperty("user.dir"));
      System.exit(1);
      }

    //read and process each record from the StudentDetails.txt
    studentTxtFile.nextLine(); //skips the header record
    studentTxtFile.useDelimiter(","); //splits variables where comma is present
    while(studentTxtFile.hasNextLine())
      {
      //passing through records from studentTxtFile to Student_Register class constructor
      studentObject.addStudent(studentTxtFile.next(), studentTxtFile.next(), studentTxtFile.next(),
studentTxtFile.next());
      studentTxtFile.nextLine();
      }
```

```
    //read and process each record from the CourseDetails.txt
    //instantiating course class and passing through records from studentTxtFile to Course class constructor
    Course courseObject = new Course(courseTxtFile.nextLine(), courseTxtFile.nextLine(),
      Double.parseDouble(courseTxtFile.nextLine()),Double.parseDouble(courseTxtFile.nextLine()),
          Double.parseDouble(courseTxtFile.nextLine()));

    //close the file
    studentTxtFile.close();
    courseTxtFile.close();

    while(true) //whilst true, while loop prints menu options to console
      {
      System.out.println("\n----------The Enrolment Register----------");
      System.out.println("----------------Menu Options----------------");
      System.out.println("1.) List Student Names");
      System.out.println("2.) Add New Student");
      System.out.println("3.) Delete Existing Student");
      System.out.println("4.) Display Specific Student Details");
      System.out.println("5.) Display Report Details");
      System.out.println("6.) Save and Exit");
      System.out.println("-----------------------------------------");
      System.out.print("Enter Value of Menu Choice: ");

      selection = scan.nextInt(); //user selects menu option as interger

      switch(selection)  //switch reads and processes user selected value
        {
        case 1:
          studentObject.listStudentNames();
```

```
            break;
        case 2:
            studentObject.createStudent();
            courseObject.updatedDetails(studentObject.maleFemalePerc(), studentObject.total());
            break;
        case 3:
            studentObject.deleteStudent();
            courseObject.updatedDetails(studentObject.maleFemalePerc(), studentObject.total());
            break;
        case 4:
            studentObject.searchStudentDetails();
            break;
        case 5:
            courseObject.printCourseDetails();
            break;
        case 6:
            System.out.println("Changes Saved. Exiting Program");
            studentObject.saveStudentDetails();
            courseObject.updatedDetails(studentObject.maleFemalePerc(),
                    studentObject.total());
            courseObject.saveCourseDetails();
            System.exit(0);
            break;
        default: //if user interger entered not found, prompt re-entry
            System.out.println("--------------------------------");
            System.out.println("Invalid Entry. Please Try Again");
            System.out.println("--------------------------------");
         break;
        }
    }
}
```

}// End of class Main

```
/*
 * This is an solution for the  program which forms the second
 * Practical Skills Assessment on COM102.
 * Created by: Niall Morrissey (B00787301), Michael Brown (B00808857)
 * Date: 18/04/2021
 * Version: 1.0
 */

import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Scanner;

/**
 * This class holds the Student_Register functionality of the program and that
 * methods that can be called from the main class user menu
 */

public class Student_Register
   {
   private Scanner scan = new Scanner(System.in); // Creates Scanner instance;
   private Student [] register;

   private int amount; // total number of students in register

   public Student_Register()
      {
      // constructor creates new Student object array 'register' for holding student details
      this.register = new Student[20];
      this.amount = 0;
      }
```

```
protected void addStudent(String name, String dob, String address, String gender)
   {
   // populates register array with new student details when method is called
   if (amount < 20) //if counter has not reached 20, populate array index at current counter int
      {
      this.register[amount] = new Student (name, dob, address, gender);
      amount++;
      }
   else //else, check array for null entries and populate where free
      {
      for (int i=0; i < 20; i++)
         {
         if (this.register[i] == null)
               {
            this.register[i] = new Student (name, dob, address, gender);
            break;
            }
         else if (this.register[i] != null && i == 19) // if no space free, state array is full
            System.out.println("Student register is full.");
         }
      }
   }


protected void createStudent()
   {
   // method to create new student entry, using while loop validation
   // so blank entries aren't accepted and gender must be 'male' or 'female'
   System.out.println("----------Enter New Student----------");
   System.out.println("Note: Blank entries are not accepted.");
   System.out.print("\nPlease enter student name: ");
   String name = scan.nextLine();
```

```
    while ("".equals(name)) {
        System.out.print("Empty entry. Please enter student name: ");
        name = scan.nextLine();
    }
    System.out.print("\nPlease enter student date of birth: ");
    String dob = scan.nextLine();
    while ("".equals(dob)) {
        System.out.print("Empty entry. Please enter student date of birth: ");
        dob = scan.nextLine();
    }
    System.out.print("\nPlease enter student address: ");
    String address = scan.nextLine();
    while ("".equals(address)) {
        System.out.print("Empty entry. Please enter student address: ");
        address = scan.nextLine();
    }
    System.out.print("\nPlease enter student gender ('male' or 'female'): ");
    String gender = scan.nextLine();
    while (!"male".equals(gender) && !"female".equals(gender)) {
        System.out.print("Empty entry. Please enter student gender: ");
        gender = scan.nextLine();
    }

    this.addStudent(name, dob, address, gender);
    }

protected void deleteStudent()
    {
    // method to delete existing student entry
    System.out.println("----------Delete Existing Student----------");
    System.out.println("Note: Select menu option 1 for list of possible names.");
```

```
   System.out.println("Note: Names entered are CASE SENSITIVE.");
   System.out.print("\nPlease enter student name to be deleted: ");
   String searchName = scan.nextLine();

   for (int i=0; i < amount; i++)
      {
      if (this.register[i] != null) // if register array entry is not null
         {
         if(this.register[i].name.equals(searchName)) // check entry for name
            {
            this.register[i] = null; // if student found, set array entry to null
            System.out.println("The name "+searchName+" has been removed");
            }
         }
      }
   }
protected void listStudentNames()
   {
    // method to list name of each student currently on course
   System.out.println("\n-----Student Names------");
   for(int i = 0; i < this.amount; i++)
      {
      if (this.register[i] != null)
         {
         this.register[i].listNames();
         }
       }
   }

protected void searchStudentDetails()
   {
```

```
   // method to search and display details of an existing student
   System.out.println("----------Search Existing Student----------");
   System.out.println("Note: Select menu option 1 for list of possible names.");
   System.out.println("Note: Names entered are CASE SENSITIVE.");
   System.out.print("\nPlease enter student name to display their details: ");
   String searchName = scan.nextLine();
   for (int i=0; i < amount; i++)
      {
      if (this.register[i] != null)
         {
         if(this.register[i].name.equals(searchName))
            {
            this.register[i].printDetails();
            }
         }
      }
   }


protected double maleFemalePerc()
   {
   // method to find male and female percentage to be returned when called
   double male = 0;
   double female = 0;
   for (int i=0; i < amount; i++)
      {
      if (this.register[i] != null)
         {
         if("male".equals(this.register[i].gender))
            male += 1;
         else
            female += 1;
```

```
        }
      }
   double total = male+female;
   double malePerc = male/total*100;

   return malePerc;
   }

 protected double total()
   {
   // method to find total number of existing students to be returned when called
   int total = 0;
   for (int i=0; i < amount; i++)
      {
      if (this.register[i] != null)
         total+=1;
      }
   return total;
   }

protected void saveStudentDetails()
   {
   // method to save and output student register array details to text file when called
   PrintWriter out = null;
   try
      {
      out = new PrintWriter("StudentDetails.txt");
      }
   catch (FileNotFoundException ex)
      {
      System.out.println(ex.getMessage());
```

```
    System.out.println("in " + System.getProperty("user.dir"));
    System.exit(1);
    }

  //process and output and  each record from the file
  out.println("Name, DOB, Address, Gender,");

  for(int i = 0; i < this.amount; i++)
    {
    if (this.register[i] != null)
      {
      out.println(this.register[i].name +","+ this.register[i].dob
      +","+ this.register[i].address +","+ this.register[i].gender +",");
      }
    }
  out.close(); // close PrintWriter class
  }
}// End of class Student_Register
```

```
/*
 * This is an solution for the  program which forms the second
 * Practical Skills Assessment on COM102.
 * Created by: Niall Morrissey (B00787301), Michael Brown (B00808857)
 * Date: 18/04/2021
 * Version: 1.0
 */

public class Student
   {
   protected String name, dob, address, gender;

   // constructor to create instance variables of each register array entry
   public Student (String name, String dob, String address, String gender)
      {
      this.name = name;
      this.dob = dob;
      this.address = address;
      this.gender = gender;
      }
   protected void listNames()
      {
      // prints out the name of student when called
      System.out.println("Student Name: " + this.name);
      }
   protected void printDetails()
      {
      // // prints out the details of a student when called
      System.out.println("\n-----Student Details------");
      System.out.println("Name: " + this.name);
      System.out.println("DOB: " + this.dob);
```

```
    System.out.println("Address: " + this.address);
    System.out.println("Gender: " + this.gender);
    }
}// End of class Student
```

```
/*
 * This is an solution for the  program which forms the second
 * Practical Skills Assessment on COM102.
 * Created by: Niall Morrissey (B00787301), Michael Brown (B00808857)
 * Date: 18/04/2021
 * Version: 1.0
 */

import java.io.FileNotFoundException;
import java.io.PrintWriter;

/**
 * This class holds the Course functionality of the program and that
 * methods that can be called from the main class user menu
 */

public class Course
   {
   protected String lecName, courseName;
   protected double malePerc, femalePerc, totalStudents;

   // constructor to create instance variables of Course details
   protected Course (String lecName, String courseName, double malePerc, double femalePerc,
double totalStudents)
      {
      this.lecName = lecName;
      this.courseName = courseName;
      this.malePerc = malePerc;
      this.femalePerc = femalePerc;
      this.totalStudents = totalStudents;
      }
```

```
protected void updatedDetails(double malePerc, double totalStudents)
   {
   // method tochange instance variable values when
   // called after student details have been altered
   this.malePerc = malePerc;
   this.femalePerc = 100-malePerc;
   this.totalStudents = totalStudents;
   }

protected void printCourseDetails()
   {
   // method to print out the details of course
   System.out.println("\n------Course Details------");
   System.out.println("Lecturer Name " + lecName);
   System.out.println("Course Name: " + courseName);
   System.out.printf("Male Percentage: " + "%.2f", malePerc);
   System.out.println("%");
   System.out.printf("Female Percentage: " + "%.2f", femalePerc);
   System.out.println("%");
   System.out.println("Total Students: " + totalStudents);
   }
protected void saveCourseDetails()
   {
   // method to save and output course details to text file when called
   PrintWriter out = null;
   try
      {
      out = new PrintWriter("CourseDetails.txt");
      }
   catch (FileNotFoundException ex)
      {
```

```
        System.out.println(ex.getMessage());
        System.out.println("in " + System.getProperty("user.dir"));
        System.exit(1);
        }

    //process and output and  each record from the file
    out.println(lecName);
    out.println(courseName);
    out.println(malePerc);
    out.println(femalePerc);
    out.println(totalStudents);

    out.close(); // close PrintWriter class
    }
}// End of class Course
```