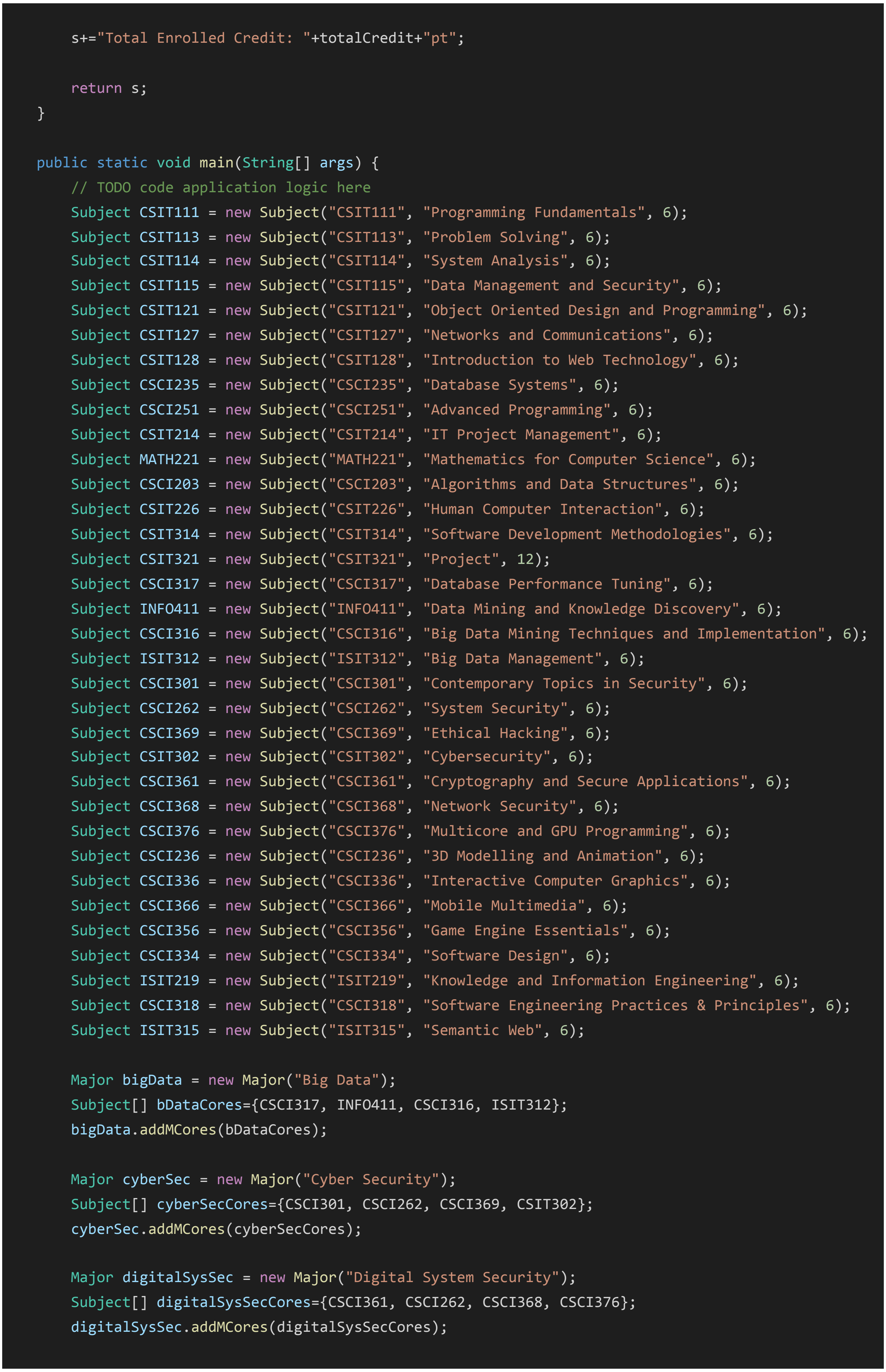
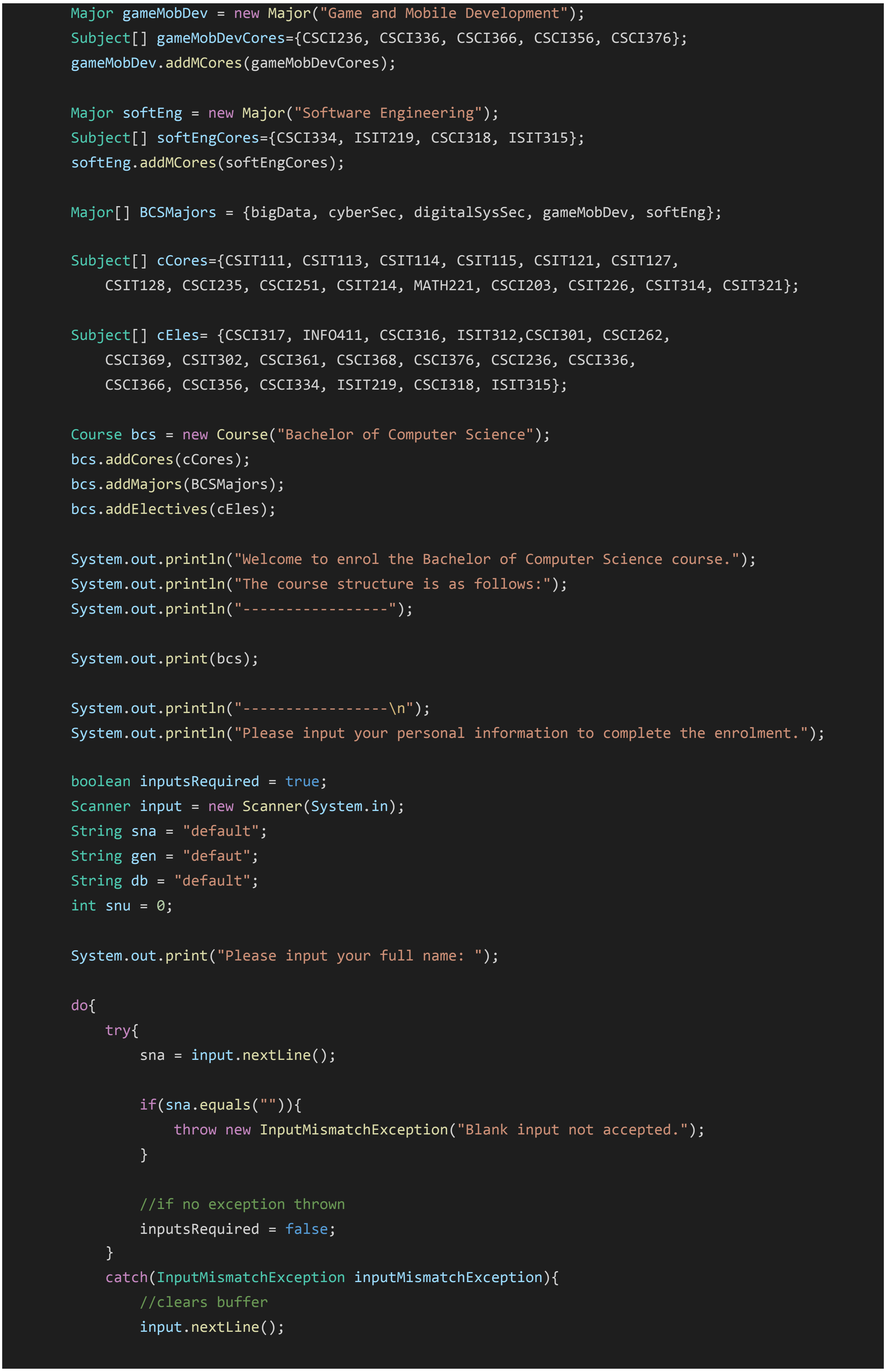
CSCI121 - Sean Overton
SN: 6421490
LAB 4:
CODE:

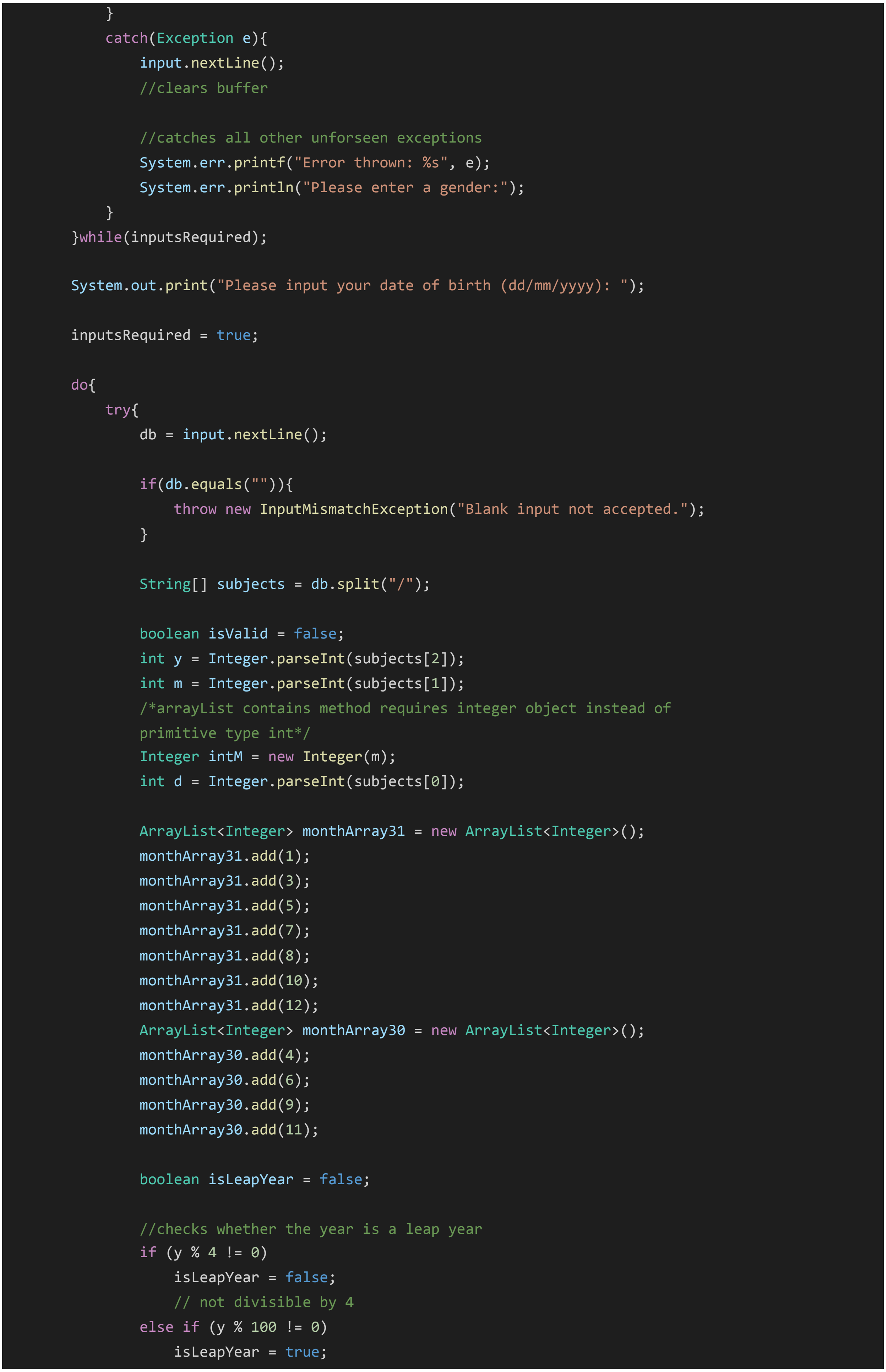
```
import java.util.ArrayList;
import java.util.Scanner;
import java.util.Arrays;
import java.util.InputMismatchException;

//custom exeption class
class InputOutOfRange extends Exception{
    //datafields
    private String range;

    //parameterised constructor
    public InputOutOfRange(String range){
        this.range = range;
    }

    public String getMessage(){
        if(range != null){
            return String.format("Input out of range. Should be between %s", getRange());
        }
        else{
            return "Input out of range.";
        }
    }

    public String getRange(){
        return this.range;
    }
}

public class Enrolment {
    private Student student;
    private String cName;
    private ArrayList<Subject> eCores;
    private Major eMajor;
    private ArrayList<Subject> eElectives;
    private int totalCredit;

    public Enrolment(Student s, String n){
        student = s;
        cName = n;
        eCores = new ArrayList<Subject>(0);
        eMajor = null;
        eElectives = new ArrayList<Subject>(0);
        totalCredit = 0;
    }

    public void enrolCores(ArrayList<Subject> c){
        eCores.addAll(c);
        for(Subject es:eCores)
            totalCredit += es.getCredit();
    }

    public void enrolMajor(Major m){
        eMajor = m;

        for(Subject es:eMajor.getMCores())
            totalCredit += es.getCredit();
    }

    public void enrolElective(Subject s){
        if(!isEnrolled(s)){
            eElectives.add(s);
            totalCredit+=s.getCredit();
        }

    }

    public int getTotalCredit(){
        return totalCredit;
    }

    public boolean isEnrolled(Subject s){
        Boolean b = false;

        b=b||eMajor.isIncluded(s);

        for(Subject es:eElectives)
            b=b||es.isSame(s);

        for(Subject ec:eCores)
            b=b||ec.isSame(s);

        return b;
    }

    public String toString(){
        String s="";

        s+=student+"\n";

        s+="Cores: \n";
        for(Subject su:eCores)
            s+=su;

        s+="\n";

        s+=eMajor;

        s+="Electives: \n";
        for(Subject su:eElectives)
            s+=su;

        s+="-----------------\n";
```



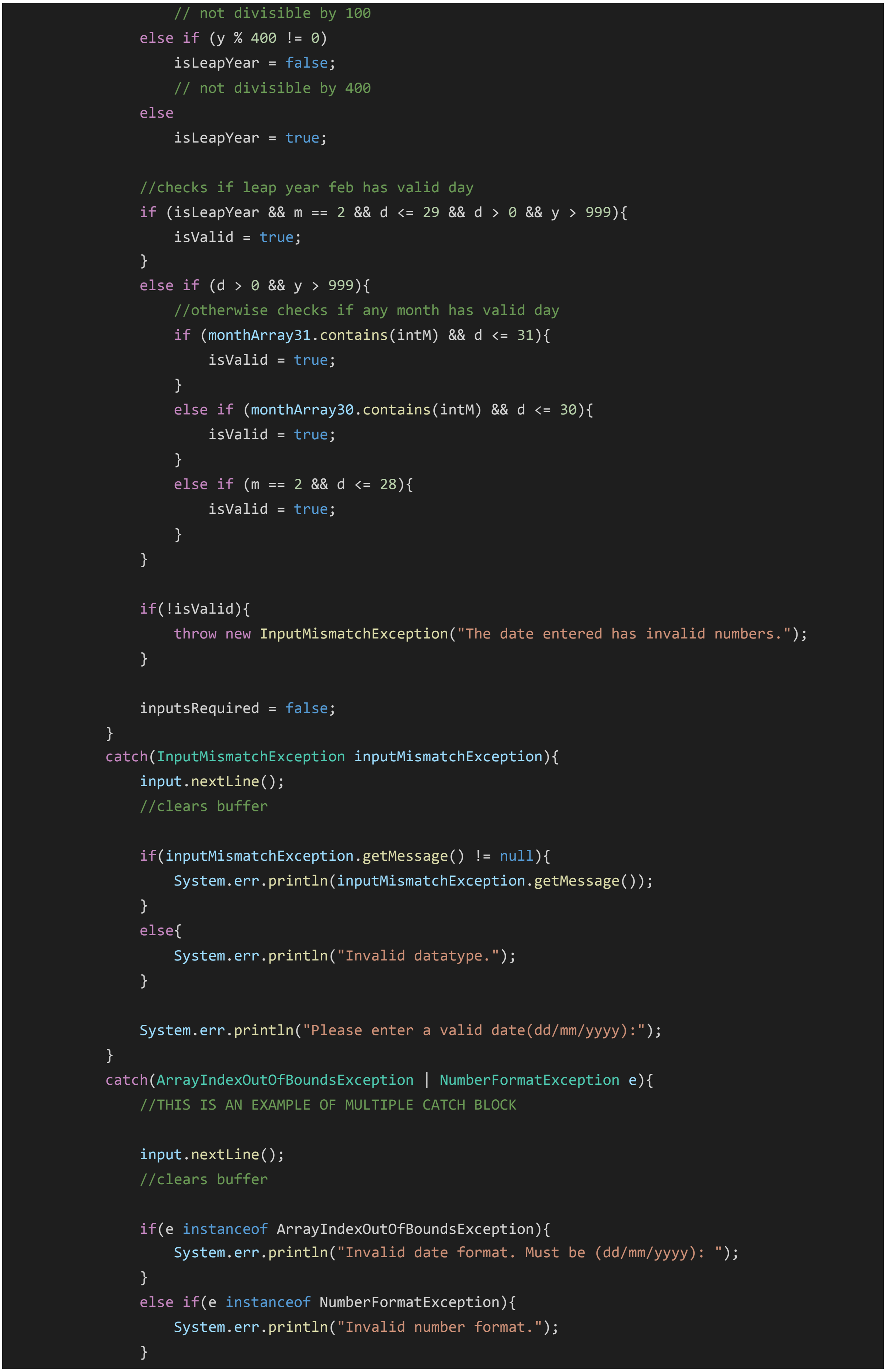
```
        s+="Total Enrolled Credit: "+totalCredit+"pt";

        return s;
    }

    public static void main(String[] args) {
        // TODO code application logic here
        Subject CSIT111 = new Subject("CSIT111", "Programming Fundamentals", 6);
        Subject CSIT113 = new Subject("CSIT113", "Problem Solving", 6);
        Subject CSIT114 = new Subject("CSIT114", "System Analysis", 6);
        Subject CSIT115 = new Subject("CSIT115", "Data Management and Security", 6);
        Subject CSIT121 = new Subject("CSIT121", "Object Oriented Design and Programming", 6);
        Subject CSIT127 = new Subject("CSIT127", "Networks and Communications", 6);
        Subject CSIT128 = new Subject("CSIT128", "Introduction to Web Technology", 6);
        Subject CSCI235 = new Subject("CSCI235", "Database Systems", 6);
        Subject CSCI251 = new Subject("CSCI251", "Advanced Programming", 6);
        Subject CSIT214 = new Subject("CSIT214", "IT Project Management", 6);
        Subject MATH221 = new Subject("MATH221", "Mathematics for Computer Science", 6);
        Subject CSCI203 = new Subject("CSCI203", "Algorithms and Data Structures", 6);
        Subject CSIT226 = new Subject("CSIT226", "Human Computer Interaction", 6);
        Subject CSIT314 = new Subject("CSIT314", "Software Development Methodologies", 6);
        Subject CSIT321 = new Subject("CSIT321", "Project", 12);
        Subject CSCI317 = new Subject("CSCI317", "Database Performance Tuning", 6);
        Subject INFO411 = new Subject("INFO411", "Data Mining and Knowledge Discovery", 6);
        Subject CSCI316 = new Subject("CSCI316", "Big Data Mining Techniques and Implementation", 6);
        Subject ISIT312 = new Subject("ISIT312", "Big Data Management", 6);
        Subject CSCI301 = new Subject("CSCI301", "Contemporary Topics in Security", 6);
        Subject CSCI262 = new Subject("CSCI262", "System Security", 6);
        Subject CSCI369 = new Subject("CSCI369", "Ethical Hacking", 6);
        Subject CSIT302 = new Subject("CSIT302", "Cybersecurity", 6);
        Subject CSCI361 = new Subject("CSCI361", "Cryptography and Secure Applications", 6);
        Subject CSCI368 = new Subject("CSCI368", "Network Security", 6);
        Subject CSCI376 = new Subject("CSCI376", "Multicore and GPU Programming", 6);
        Subject CSCI236 = new Subject("CSCI236", "3D Modelling and Animation", 6);
        Subject CSCI336 = new Subject("CSCI336", "Interactive Computer Graphics", 6);
        Subject CSCI366 = new Subject("CSCI366", "Mobile Multimedia", 6);
        Subject CSCI356 = new Subject("CSCI356", "Game Engine Essentials", 6);
        Subject CSCI334 = new Subject("CSCI334", "Software Design", 6);
        Subject ISIT219 = new Subject("ISIT219", "Knowledge and Information Engineering", 6);
        Subject CSCI318 = new Subject("CSCI318", "Software Engineering Practices & Principles", 6);
        Subject ISIT315 = new Subject("ISIT315", "Semantic Web", 6);

        Major bigData = new Major("Big Data");
        Subject[] bDataCores={CSCI317, INFO411, CSCI316, ISIT312};
        bigData.addMCores(bDataCores);

        Major cyberSec = new Major("Cyber Security");
        Subject[] cyberSecCores={CSCI301, CSCI262, CSCI369, CSIT302};
        cyberSec.addMCores(cyberSecCores);

        Major digitalSysSec = new Major("Digital System Security");
        Subject[] digitalSysSecCores={CSCI361, CSCI262, CSCI368, CSCI376};
        digitalSysSec.addMCores(digitalSysSecCores);
```



```java
        Major gameMobDev = new Major("Game and Mobile Development");
        Subject[] gameMobDevCores={CSCI236, CSCI336, CSCI366, CSCI356, CSCI376};
        gameMobDev.addMCores(gameMobDevCores);

        Major softEng = new Major("Software Engineering");
        Subject[] softEngCores={CSCI334, ISIT219, CSCI318, ISIT315};
        softEng.addMCores(softEngCores);

        Major[] BCSMajors = {bigData, cyberSec, digitalSysSec, gameMobDev, softEng};

        Subject[] cCores={CSIT111, CSIT113, CSIT114, CSIT115, CSIT121, CSIT127,
            CSIT128, CSCI235, CSCI251, CSIT214, MATH221, CSCI203, CSIT226, CSIT314, CSIT321};

        Subject[] cEles= {CSCI317, INFO411, CSCI316, ISIT312,CSCI301, CSCI262,
            CSCI369, CSIT302, CSCI361, CSCI368, CSCI376, CSCI236, CSCI336,
            CSCI366, CSCI356, CSCI334, ISIT219, CSCI318, ISIT315};

        Course bcs = new Course("Bachelor of Computer Science");
        bcs.addCores(cCores);
        bcs.addMajors(BCSMajors);
        bcs.addElectives(cEles);

        System.out.println("Welcome to enrol the Bachelor of Computer Science course.");
        System.out.println("The course structure is as follows:");
        System.out.println("-----------------");

        System.out.print(bcs);

        System.out.println("-----------------\n");
        System.out.println("Please input your personal information to complete the enrolment.");

        boolean inputsRequired = true;
        Scanner input = new Scanner(System.in);
        String sna = "default";
        String gen = "defaut";
        String db = "default";
        int snu = 0;

        System.out.print("Please input your full name: ");

        do{
            try{
                sna = input.nextLine();

                if(sna.equals("")){
                    throw new InputMismatchException("Blank input not accepted.");
                }

                //if no exception thrown
                inputsRequired = false;
            }
            catch(InputMismatchException inputMismatchException){
                //clears buffer
                input.nextLine();
                if(inputMismatchException.getMessage() != null){
                    System.err.println(inputMismatchException.getMessage());
                }
                else{
                    System.err.println("Invalid datatype.");
                }
                System.err.println("Please enter valid full-name:");
            }
            catch(Exception e){
                //clears buffer
                input.nextLine();

                //catches all other unforseen exceptions
                System.err.printf("Error thrown: %s", e);
                System.err.println("Please enter valid full-name:");
            }
        }while(inputsRequired);

        inputsRequired = true;

        System.out.print("Please input your student number: ");

        String number = "";

        do{
            try{
                number = input.nextLine();
                snu = Integer.parseInt(number);

                if(snu < 10000 || snu >=100000){
                    throw new InputOutOfRange("10000-99999");
                }
                //if no exception thrown
                inputsRequired = false;
            }
            catch(NumberFormatException e){
                //clears buffer
                input.nextLine();

                if(number.equals("")){
                    System.err.println("Blank input not accepted.");
                }

                System.err.println("Invalid datatype.");
                System.err.println("Please enter valid integer input.");
            }
            catch(InputMismatchException e){
                //clears buffer
                input.nextLine();

                if(e.getMessage() != null){
                    System.err.println(e.getMessage());
                }
                else{
                    System.err.println("Invalid datatype.");
                }
                System.err.println("Please enter valid Student number:");
            }
            catch(InputOutOfRange inputOutOfRange){
                //clears buffer
                input.nextLine();

                System.err.println(inputOutOfRange.getMessage());
                System.err.println("Please enter valid Student number:");
            }
            catch(Exception e){
                //clears buffer
                input.nextLine();

                //catches all other unforseen exceptions
                System.err.printf("Error thrown: %s", e);
                System.err.println("Please enter valid Student number:");
            }
        }while(inputsRequired);

        inputsRequired = true;

        System.out.print("Please input your gender: ");

        do{
            try{
                gen = input.nextLine();

                if(gen.equals("")){
                    throw new InputMismatchException("Blank input not accepted.");
                }

                //asertion used to enforce input constraints
                gen = gen.toLowerCase();

                assert(gen.equals("male") || gen.equals("female")) : "Please enter a valid gender. Not
" + gen;

                inputsRequired = false;
            }
            catch(AssertionError assertionError){
                System.err.printf("Assertion error caught. Please enter 'male' OR 'female'. \n%s: ",
assertionError.getMessage());
            }
            catch(InputMismatchException inputMismatchException){
                input.nextLine();
                //clears buffer

                if(inputMismatchException.getMessage() != null){
                    System.err.println(inputMismatchException.getMessage());
                }
                else{
                    System.err.println("Invalid datatype.");
                }
                System.err.println("Please enter a gender:");
```



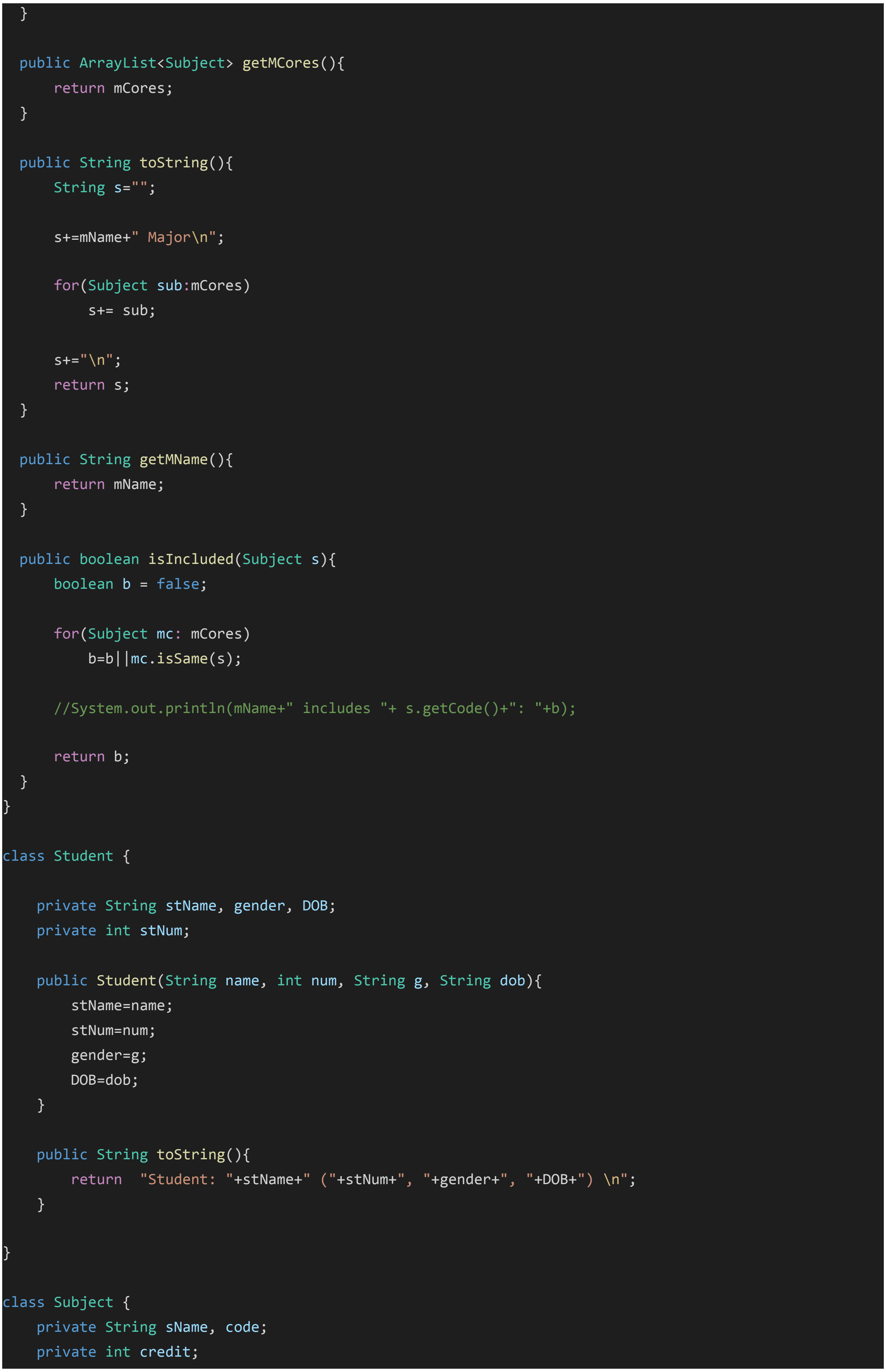
```
            }
            catch(Exception e){
                input.nextLine();
                //clears buffer

                //catches all other unforseen exceptions
                System.err.printf("Error thrown: %s", e);
                System.err.println("Please enter a gender:");
            }
        }while(inputsRequired);

        System.out.print("Please input your date of birth (dd/mm/yyyy): ");

        inputsRequired = true;

        do{
            try{
                db = input.nextLine();

                if(db.equals("")){
                    throw new InputMismatchException("Blank input not accepted.");
                }

                String[] subjects = db.split("/");

                boolean isValid = false;
                int y = Integer.parseInt(subjects[2]);
                int m = Integer.parseInt(subjects[1]);
                /*arrayList contains method requires integer object instead of
                primitive type int*/
                Integer intM = new Integer(m);
                int d = Integer.parseInt(subjects[0]);

                ArrayList<Integer> monthArray31 = new ArrayList<Integer>();
                monthArray31.add(1);
                monthArray31.add(3);
                monthArray31.add(5);
                monthArray31.add(7);
                monthArray31.add(8);
                monthArray31.add(10);
                monthArray31.add(12);
                ArrayList<Integer> monthArray30 = new ArrayList<Integer>();
                monthArray30.add(4);
                monthArray30.add(6);
                monthArray30.add(9);
                monthArray30.add(11);

                boolean isLeapYear = false;

                //checks whether the year is a leap year
                if (y % 4 != 0)
                    isLeapYear = false;
                    // not divisible by 4
                else if (y % 100 != 0)
                    isLeapYear = true;
```



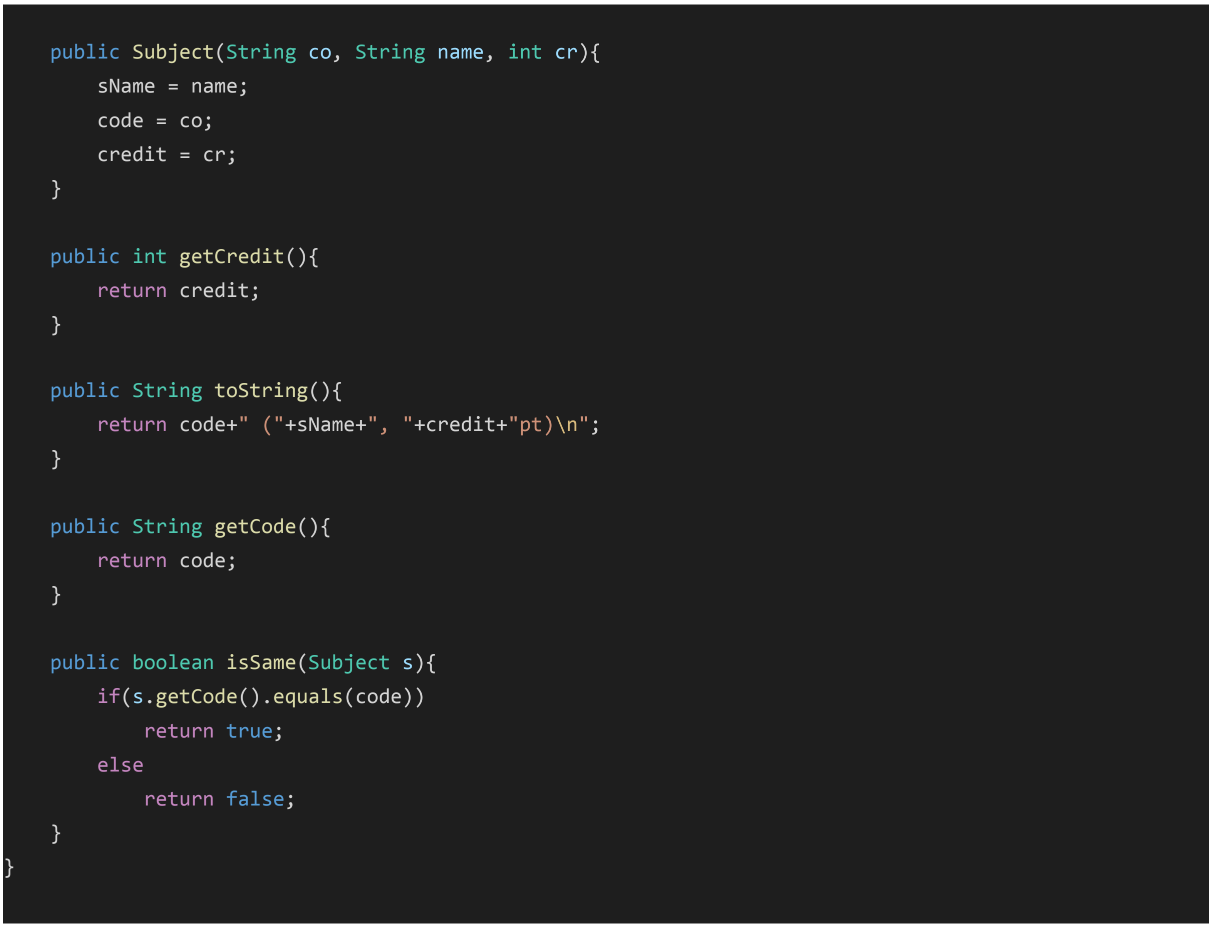
```java
                // not divisible by 100
            else if (y % 400 != 0)
                isLeapYear = false;
                // not divisible by 400
            else
                isLeapYear = true;

            //checks if leap year feb has valid day
            if (isLeapYear && m == 2 && d <= 29 && d > 0 && y > 999){
                isValid = true;
            }
            else if (d > 0 && y > 999){
                //otherwise checks if any month has valid day
                if (monthArray31.contains(intM) && d <= 31){
                    isValid = true;
                }
                else if (monthArray30.contains(intM) && d <= 30){
                    isValid = true;
                }
                else if (m == 2 && d <= 28){
                    isValid = true;
                }
            }

            if(!isValid){
                throw new InputMismatchException("The date entered has invalid numbers.");
            }

            inputsRequired = false;
        }
        catch(InputMismatchException inputMismatchException){
            input.nextLine();
            //clears buffer

            if(inputMismatchException.getMessage() != null){
                System.err.println(inputMismatchException.getMessage());
            }
            else{
                System.err.println("Invalid datatype.");
            }

            System.err.println("Please enter a valid date(dd/mm/yyyy):");
        }
        catch(ArrayIndexOutOfBoundsException | NumberFormatException e){
            //THIS IS AN EXAMPLE OF MULTIPLE CATCH BLOCK

            input.nextLine();
            //clears buffer

            if(e instanceof ArrayIndexOutOfBoundsException){
                System.err.println("Invalid date format. Must be (dd/mm/yyyy): ");
            }
            else if(e instanceof NumberFormatException){
                System.err.println("Invalid number format.");
            }
```

```
                System.err.println("Please enter a valid date(dd/mm/yyyy):");
            }
            catch(Exception e){
                input.nextLine();
                //clears buffer

                //catches all other unforseen exceptions
                System.err.printf("Error thrown: %s\n", e);
                System.err.println("Please enter a valid date(dd/mm/yyyy):");
            }
        }while(inputsRequired);

        //creates student object and enrolment object
        Student std = new Student(sna, snu, gen, db);

        Enrolment enrol = new Enrolment(std, bcs.getCName());

        //automatically enrols into core subjects
        enrol.enrolCores(bcs.getCores());

        System.out.println("\nThanks for your information, we have enrolled you into the Bachelor of Computer Science course.");
        System.out.println("In order to complete the enrolment, please select a major from the list.");
        for (int i = 1; i< bcs.getMajors().size()+1; i++)
            System.out.println(i+": "+bcs.getMajors().get(i-1).getMName());

        System.out.print("\nPlease input the index number before the major:");

        inputsRequired = true;

        /*required so it is initialised but in reality won't get past
        do/while loop that will guarruntee a valid user input */
        int m = 1;

        do{
            try{
                number = input.nextLine();
                m = Integer.parseInt(number);

                if(number.equals("")){
                    throw new InputMismatchException("Blank input not accepted.");
                }

                if(m <= 0 || m > 5){
                    throw new InputOutOfRange("1-5");
                }

                //if no exception thrown
                inputsRequired = false;
            }
            catch(InputOutOfRange inputOutOfRange){
                //clears buffer
                input.nextLine();
```

```java
                System.err.println(inputOutOfRange.getMessage());
                System.err.println("Please enter valid integer input.");
            }
            catch(NumberFormatException e){
                //clears buffer
                input.nextLine();

                System.err.println("Invalid datatype.");
                System.err.println("Please enter valid integer input.");
            }
            catch(InputMismatchException e){
                //clears buffer
                input.nextLine();

                if(e.getMessage() != null){
                    System.err.println(e.getMessage());
                }
                else{
                    System.err.println("Invalid datatype.");
                }

                System.err.println("Please enter valid integer input.");
            }
            catch(Exception e){
                //catches all other unforseen exceptions
                System.err.printf("Error thrown: %s", e);
                System.err.println("Please enter valid integer input.");
            }
        }while(inputsRequired);

        System.out.println();

        enrol.enrolMajor(bcs.getMajors().get(m-1));

        System.out.println("You select the " + bcs.getMajors().get(m-1).getMName() + " major.");
        System.out.print(bcs.getMajors().get(m-1));

        System.out.println("In order to complete the enrolment, please select elective subjects from the list.\n");
        System.out.println("Electives:");
        for(Subject es:bcs.getElectives())
                System.out.print(es);

        System.out.println();

        while(enrol.getTotalCredit()<bcs.getCCredit()){
            int n = (bcs.getCCredit()-enrol.getTotalCredit())/6;

            //REQUIRED FOR EXCEPTION HANDLING
            boolean matched = false;

            System.out.print("Please select "+n+" more elective subjects by inputing the subject codes (seperate by comma and space):");
            try{
                matched = false;
                String[] subs = input.nextLine().split(", ");

                for(String s:subs){
                    matched = false;

                    if(s.equals("")){
                        throw new Exception("Blank input not accepted.");
                    }

                    for(Subject es:bcs.getElectives()){
                        if(es.getCode().equals(s)){
                            //checks if already enrolled
                            if(enrol.isEnrolled(es)){
                                throw new Exception("Already enrolled in subject.");
                            }
                            else{
                                //otherwise enrols
                                enrol.enrolElective(es);
                                matched = true;
                                break;
                            }
                        }
                    }

                    if(!matched){
                        throw new InputMismatchException(s);
                    }
                }
            }
            catch(InputMismatchException inputMismatchException){
                //clear buffer
                input.nextLine();

                if(inputMismatchException.getMessage() !=null){
                    System.err.printf("%s is an invalid subject code.\n", 
inputMismatchException.getMessage());
                }
                else{
                    System.err.println("Invalid data type input.");
                }
                System.err.println("Please enter only valid subject codes. NOTE: if multiple must be 
seperated by ', '");
            }
            catch(ArrayIndexOutOfBoundsException e){
                //clear buffer
                input.nextLine();

                System.err.println("Invalid subject code input: ");
                System.err.println("Please enter only valid subject codes. NOTE: if multiple must be 
seperated by ', '");
            }
            catch(Exception e){
                //clears buffer
                input.nextLine();
                if(e.getMessage() !=null){
                    System.err.println(e.getMessage());
                }
                else{
                    System.err.printf("Error thrown: %s", e);
                }

                System.err.println("Please enter only valid subject codes. NOTE: if multiple must be
seperated by ', '");
            }
            finally{
                System.out.println("NOTICE: Enrolment almost complete! (this is a finally clause)");
            }
        }

        System.out.println("\nCongratulatoins. You had completed the enrolment to
"+bcs.getCName()+".");
        System.out.println("--------------------");
        System.out.println(enrol);
    }

}

class Course {
    private String cName;
    private ArrayList<Subject> cores;
    private ArrayList<Major> majors;
    private ArrayList<Subject> electives;
    private int cCredit;

    public Course(String n){
        cName = n;
        cCredit = 144;
        cores = new ArrayList<Subject>(0);
        majors = new ArrayList<Major>(0);
        electives = new ArrayList<Subject>(0);
    }

    public void addCores(Subject[] co){
        cores.addAll(Arrays.asList(co));
    }

    public void addMajors(Major[] ma){
        majors.addAll(Arrays.asList(ma));
    }

    public void addElectives(Subject[] el){
        electives.addAll(Arrays.asList(el));
    }

    public ArrayList<Subject> getCores(){
        return cores;
    }

    public ArrayList<Subject> getElectives(){
        return electives;
    }

    public String toString(){
        String s="Course: ";

        s+=cName+"\n\n";

        s+="Cores: \n";
        for(Subject su:cores)
            s+=su;

        s+="\n";

        for(Major m:majors)
            s+=m;

        s+="Electives: \n";

        for(Subject su:electives)
            s+=su;

        return s;
    }

    public String getCName(){
        return cName;
    }

    public ArrayList<Major> getMajors(){
        return majors;
    }

    public int getCCredit(){
        return cCredit;
    }
}

enum MajorName{
    BigData, CyberSecurity, DigitialSystemsSecurity,
    GameandMobileDevelopment, SoftwareEngineering
}

class Major {
  private String mName;
  private ArrayList<Subject> mCores;


  public Major(String n){
      mName = n;
      mCores = new ArrayList<Subject>(0);
  }

  public void addMCores(Subject[] cores){
      mCores.addAll(Arrays.asList(cores));
```

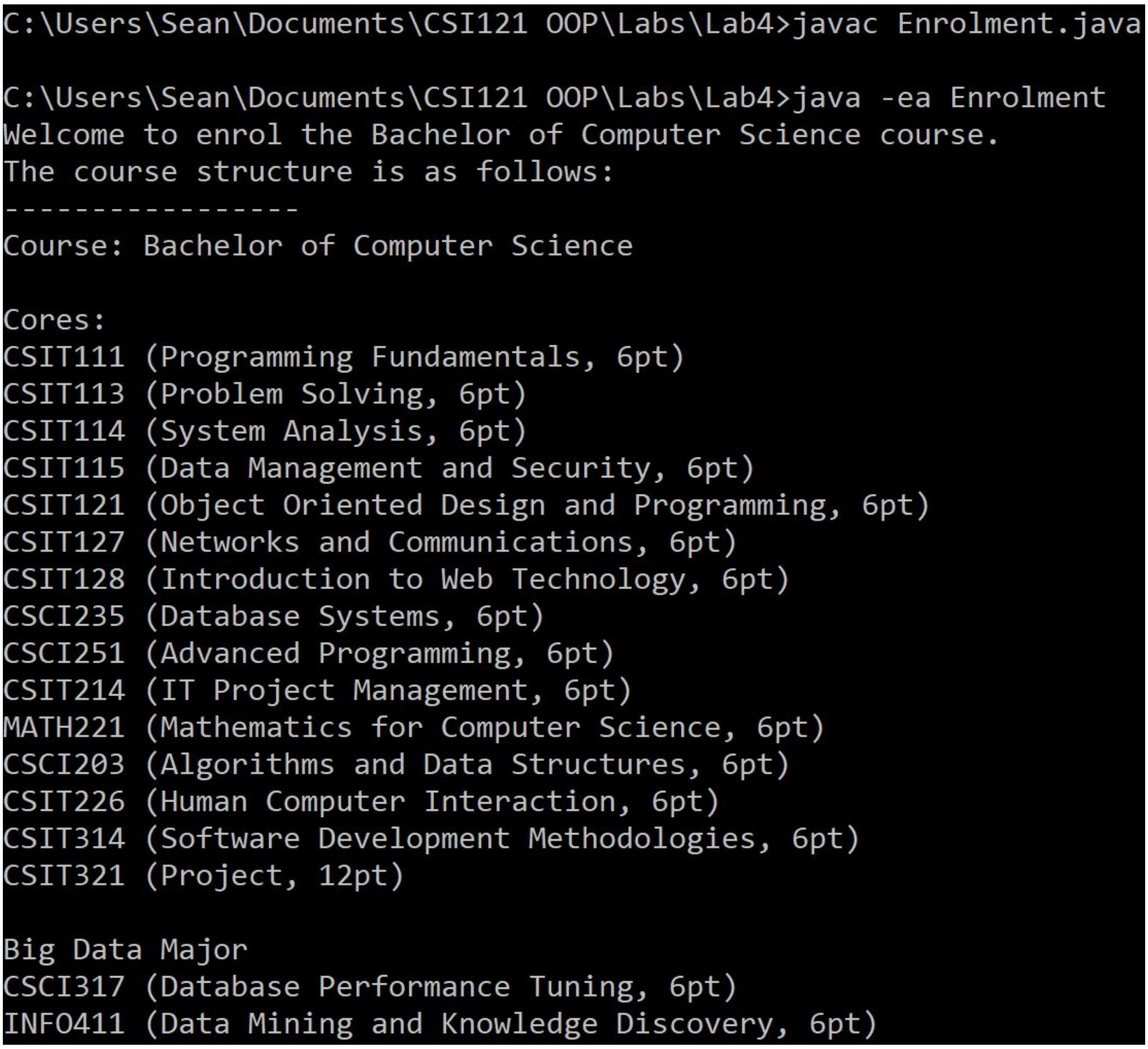


```
  }

  public ArrayList<Subject> getMCores(){
      return mCores;
  }

  public String toString(){
      String s="";

      s+=mName+" Major\n";

      for(Subject sub:mCores)
          s+= sub;

      s+="\n";
      return s;
  }

  public String getMName(){
      return mName;
  }

  public boolean isIncluded(Subject s){
      boolean b = false;

      for(Subject mc: mCores)
          b=b||mc.isSame(s);

      //System.out.println(mName+" includes "+ s.getCode()+": "+b);

      return b;
  }
}

class Student {

    private String stName, gender, DOB;
    private int stNum;

    public Student(String name, int num, String g, String dob){
        stName=name;
        stNum=num;
        gender=g;
        DOB=dob;
    }

    public String toString(){
        return  "Student: "+stName+" ("+stNum+", "+gender+", "+DOB+") \n";
    }

}

class Subject {
    private String sName, code;
    private int credit;
```

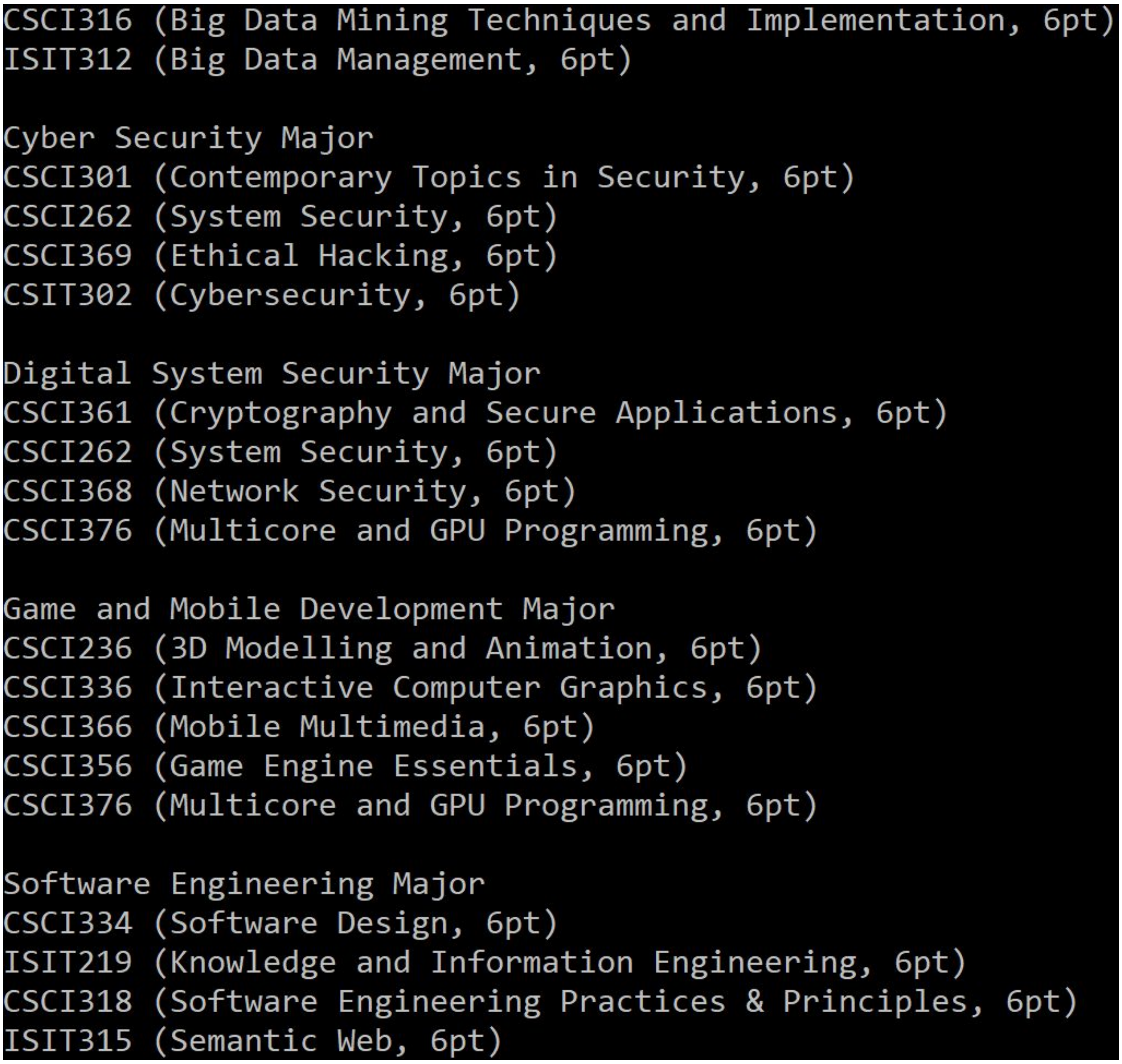


```
    public Subject(String co, String name, int cr){
        sName = name;
        code = co;
        credit = cr;
    }

    public int getCredit(){
        return credit;
    }

    public String toString(){
        return code+" ("+sName+", "+credit+"pt)\n";
    }

    public String getCode(){
        return code;
    }

    public boolean isSame(Subject s){
        if(s.getCode().equals(code))
            return true;
        else
            return false;
    }
}
```

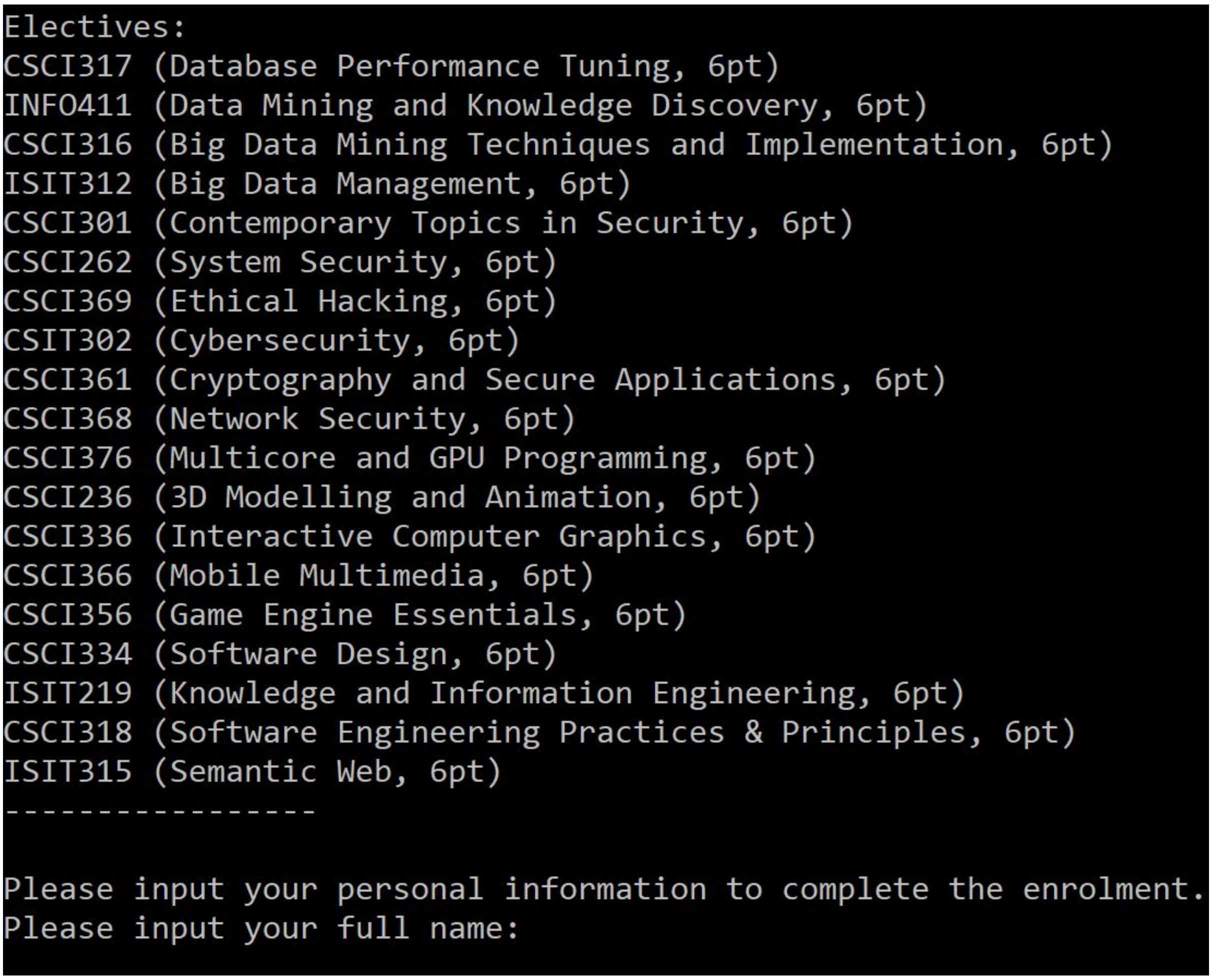
SCREENSHOTS TESTING COMPILATION AND INVALID INPUTS:



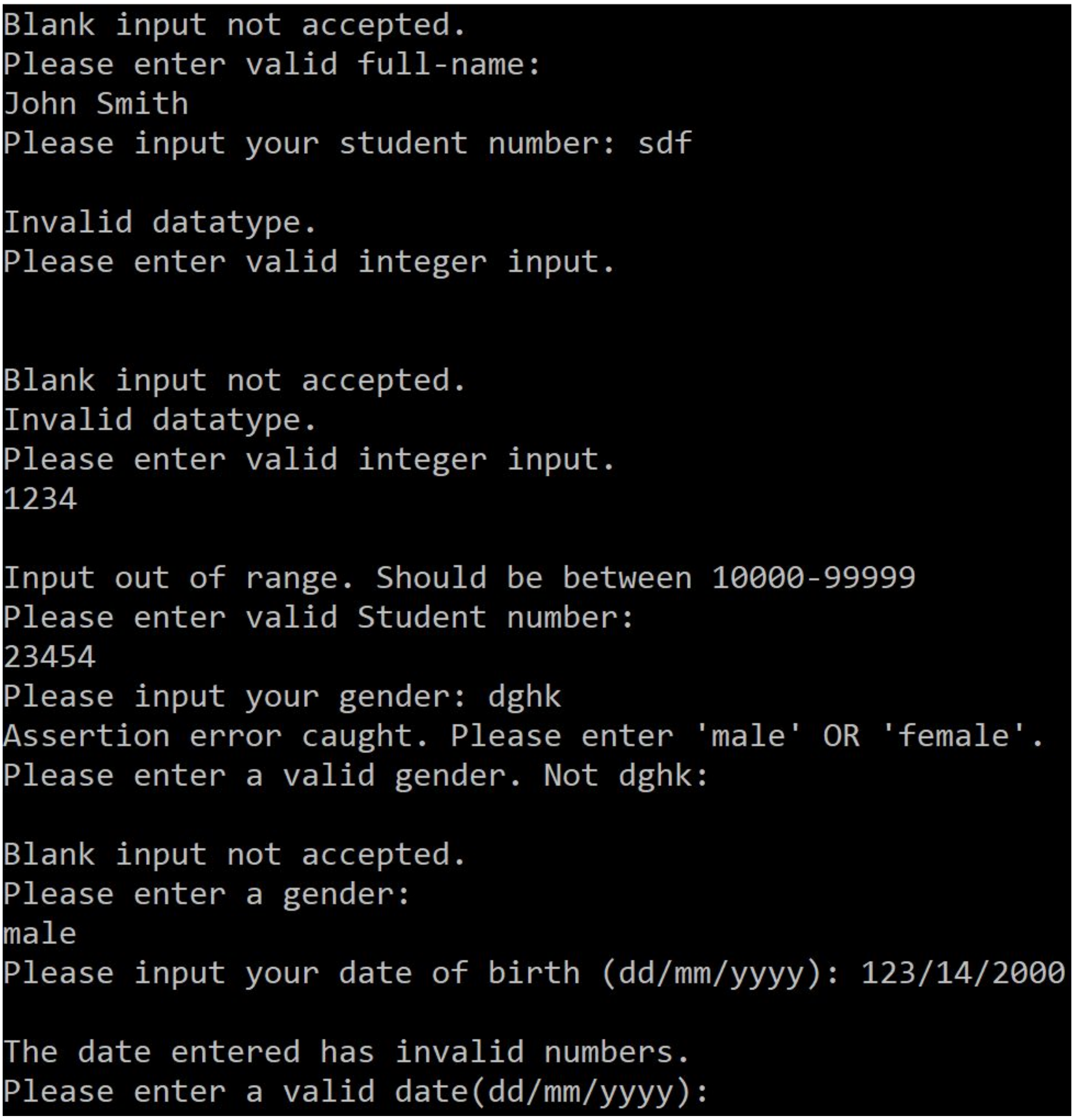
```
C:\Users\Sean\Documents\CSI121 OOP\Labs\Lab4>javac Enrolment.java

C:\Users\Sean\Documents\CSI121 OOP\Labs\Lab4>java -ea Enrolment
Welcome to enrol the Bachelor of Computer Science course.
The course structure is as follows:
-----------------
Course: Bachelor of Computer Science

Cores:
CSIT111 (Programming Fundamentals, 6pt)
CSIT113 (Problem Solving, 6pt)
CSIT114 (System Analysis, 6pt)
CSIT115 (Data Management and Security, 6pt)
CSIT121 (Object Oriented Design and Programming, 6pt)
CSIT127 (Networks and Communications, 6pt)
CSIT128 (Introduction to Web Technology, 6pt)
CSCI235 (Database Systems, 6pt)
CSCI251 (Advanced Programming, 6pt)
CSIT214 (IT Project Management, 6pt)
MATH221 (Mathematics for Computer Science, 6pt)
CSCI203 (Algorithms and Data Structures, 6pt)
CSIT226 (Human Computer Interaction, 6pt)
CSIT314 (Software Development Methodologies, 6pt)
CSIT321 (Project, 12pt)

Big Data Major
CSCI317 (Database Performance Tuning, 6pt)
INFO411 (Data Mining and Knowledge Discovery, 6pt)
```



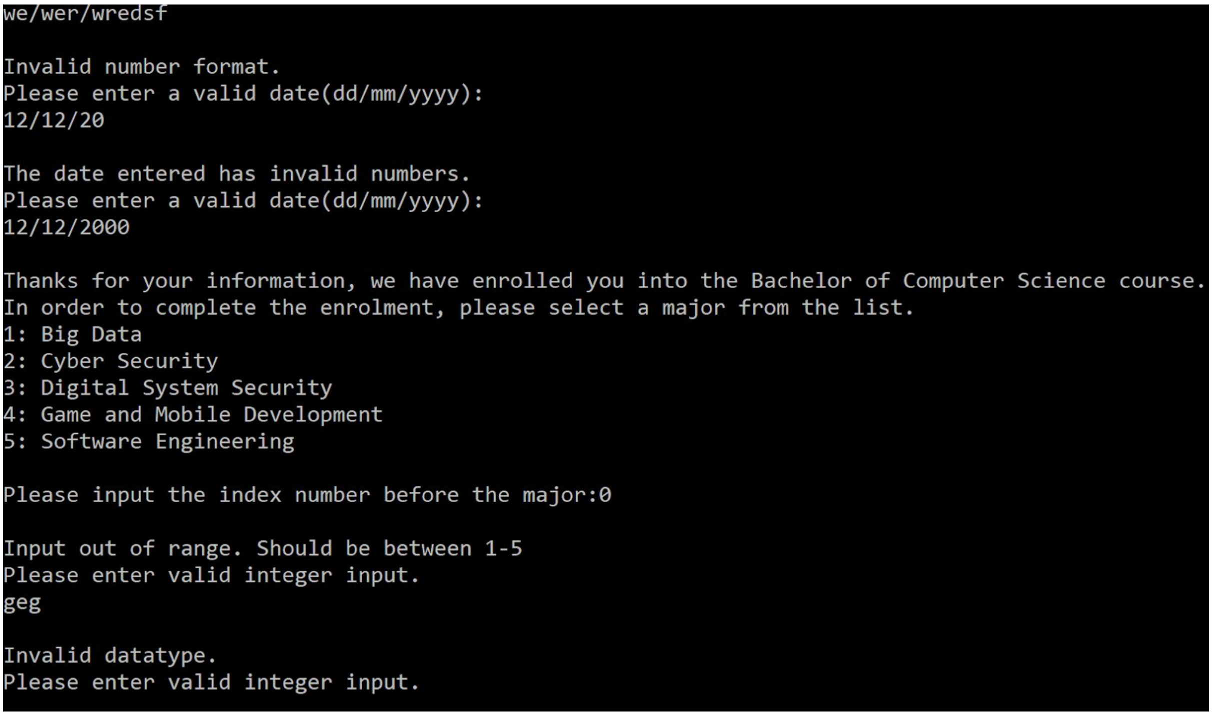
```
CSCI316 (Big Data Mining Techniques and Implementation, 6pt)
ISIT312 (Big Data Management, 6pt)

Cyber Security Major
CSCI301 (Contemporary Topics in Security, 6pt)
CSCI262 (System Security, 6pt)
CSCI369 (Ethical Hacking, 6pt)
CSIT302 (Cybersecurity, 6pt)

Digital System Security Major
CSCI361 (Cryptography and Secure Applications, 6pt)
CSCI262 (System Security, 6pt)
CSCI368 (Network Security, 6pt)
CSCI376 (Multicore and GPU Programming, 6pt)

Game and Mobile Development Major
CSCI236 (3D Modelling and Animation, 6pt)
CSCI336 (Interactive Computer Graphics, 6pt)
CSCI366 (Mobile Multimedia, 6pt)
CSCI356 (Game Engine Essentials, 6pt)
CSCI376 (Multicore and GPU Programming, 6pt)

Software Engineering Major
CSCI334 (Software Design, 6pt)
ISIT219 (Knowledge and Information Engineering, 6pt)
CSCI318 (Software Engineering Practices & Principles, 6pt)
ISIT315 (Semantic Web, 6pt)
```



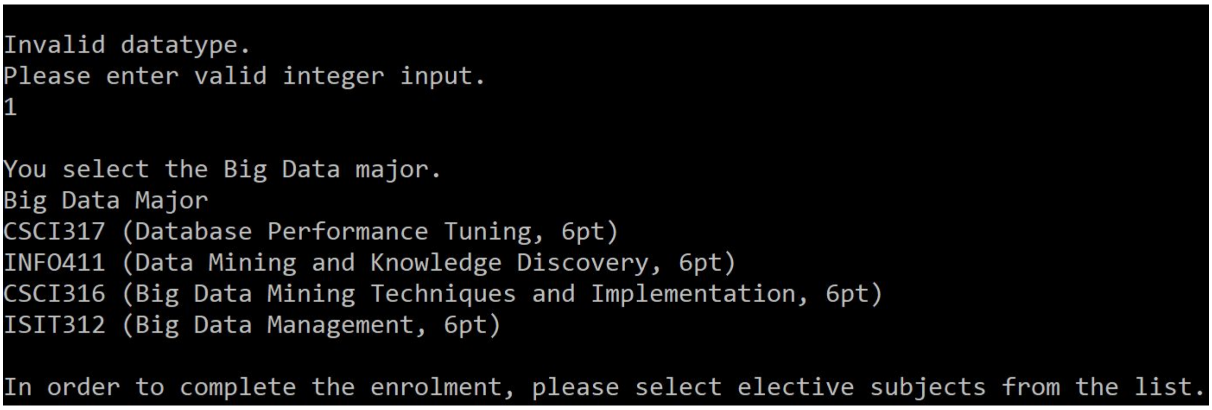
```
Electives:
CSCI317 (Database Performance Tuning, 6pt)
INFO411 (Data Mining and Knowledge Discovery, 6pt)
CSCI316 (Big Data Mining Techniques and Implementation, 6pt)
ISIT312 (Big Data Management, 6pt)
CSCI301 (Contemporary Topics in Security, 6pt)
CSCI262 (System Security, 6pt)
CSCI369 (Ethical Hacking, 6pt)
CSIT302 (Cybersecurity, 6pt)
CSCI361 (Cryptography and Secure Applications, 6pt)
CSCI368 (Network Security, 6pt)
CSCI376 (Multicore and GPU Programming, 6pt)
CSCI236 (3D Modelling and Animation, 6pt)
CSCI336 (Interactive Computer Graphics, 6pt)
CSCI366 (Mobile Multimedia, 6pt)
CSCI356 (Game Engine Essentials, 6pt)
CSCI334 (Software Design, 6pt)
ISIT219 (Knowledge and Information Engineering, 6pt)
CSCI318 (Software Engineering Practices & Principles, 6pt)
ISIT315 (Semantic Web, 6pt)
-----------------

Please input your personal information to complete the enrolment.
Please input your full name:
```



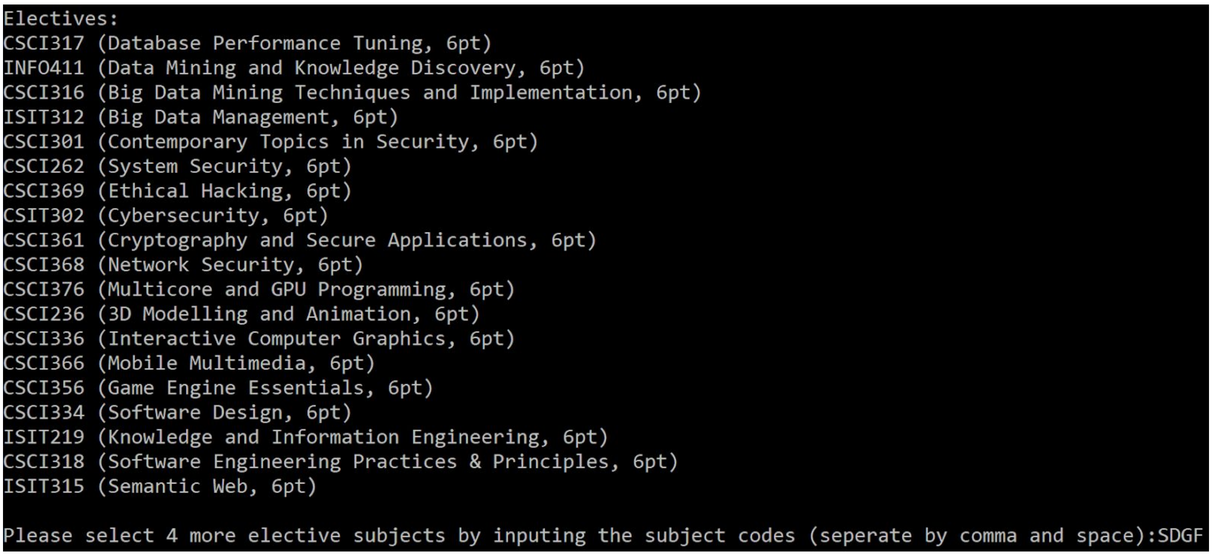
```
Blank input not accepted.
Please enter valid full-name:
John Smith
Please input your student number: sdf

Invalid datatype.
Please enter valid integer input.


Blank input not accepted.
Invalid datatype.
Please enter valid integer input.
1234

Input out of range. Should be between 10000-99999
Please enter valid Student number:
23454
Please input your gender: dghk
Assertion error caught. Please enter 'male' OR 'female'.
Please enter a valid gender. Not dghk:

Blank input not accepted.
Please enter a gender:
male
Please input your date of birth (dd/mm/yyyy): 123/14/2000

The date entered has invalid numbers.
Please enter a valid date(dd/mm/yyyy):
```



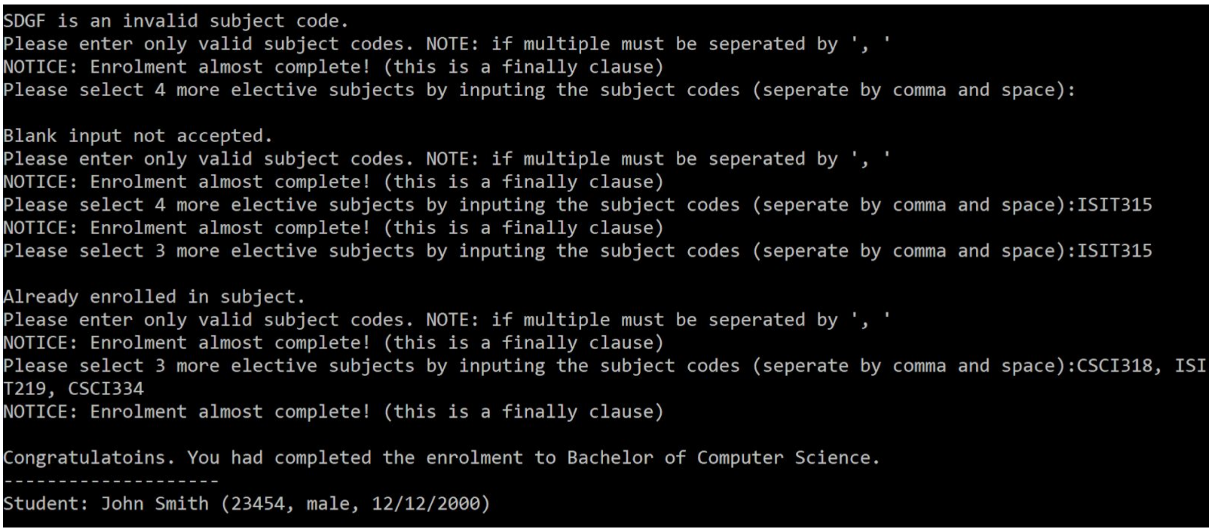
```
we/wer/wredsf

Invalid number format.
Please enter a valid date(dd/mm/yyyy):
12/12/20

The date entered has invalid numbers.
Please enter a valid date(dd/mm/yyyy):
12/12/2000

Thanks for your information, we have enrolled you into the Bachelor of Computer Science course.
In order to complete the enrolment, please select a major from the list.
1: Big Data
2: Cyber Security
3: Digital System Security
4: Game and Mobile Development
5: Software Engineering

Please input the index number before the major:0

Input out of range. Should be between 1-5
Please enter valid integer input.
geg

Invalid datatype.
Please enter valid integer input.
```



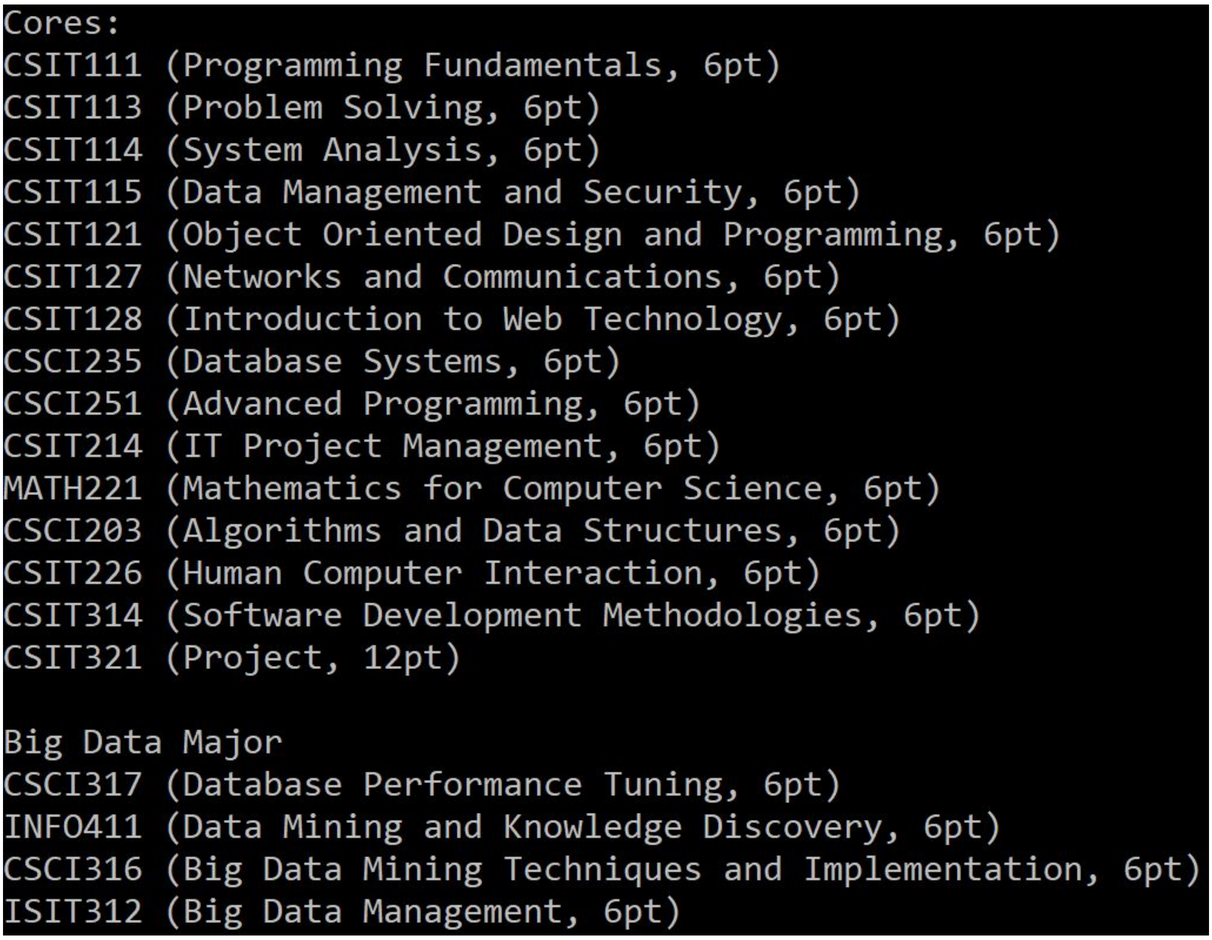
```
Invalid datatype.
Please enter valid integer input.
1

You select the Big Data major.
Big Data Major
CSCI317 (Database Performance Tuning, 6pt)
INFO411 (Data Mining and Knowledge Discovery, 6pt)
CSCI316 (Big Data Mining Techniques and Implementation, 6pt)
ISIT312 (Big Data Management, 6pt)

In order to complete the enrolment, please select elective subjects from the list.
```



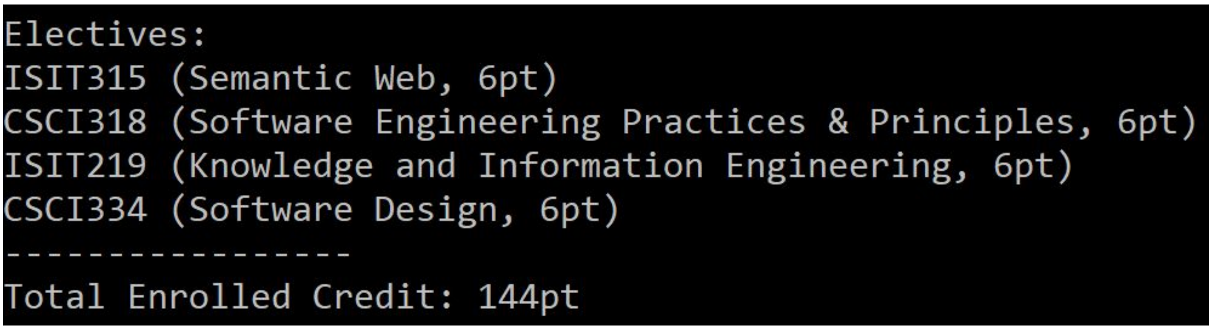
```
Electives:
CSCI317 (Database Performance Tuning, 6pt)
INFO411 (Data Mining and Knowledge Discovery, 6pt)
CSCI316 (Big Data Mining Techniques and Implementation, 6pt)
ISIT312 (Big Data Management, 6pt)
CSCI301 (Contemporary Topics in Security, 6pt)
CSCI262 (System Security, 6pt)
CSCI369 (Ethical Hacking, 6pt)
CSIT302 (Cybersecurity, 6pt)
CSCI361 (Cryptography and Secure Applications, 6pt)
CSCI368 (Network Security, 6pt)
CSCI376 (Multicore and GPU Programming, 6pt)
CSCI236 (3D Modelling and Animation, 6pt)
CSCI336 (Interactive Computer Graphics, 6pt)
CSCI366 (Mobile Multimedia, 6pt)
CSCI356 (Game Engine Essentials, 6pt)
CSCI334 (Software Design, 6pt)
ISIT219 (Knowledge and Information Engineering, 6pt)
CSCI318 (Software Engineering Practices & Principles, 6pt)
ISIT315 (Semantic Web, 6pt)

Please select 4 more elective subjects by inputing the subject codes (seperate by comma and space):SDGF
```



```
SDGF is an invalid subject code.
Please enter only valid subject codes. NOTE: if multiple must be seperated by ', '
NOTICE: Enrolment almost complete! (this is a finally clause)
Please select 4 more elective subjects by inputing the subject codes (seperate by comma and space):

Blank input not accepted.
Please enter only valid subject codes. NOTE: if multiple must be seperated by ', '
NOTICE: Enrolment almost complete! (this is a finally clause)
Please select 4 more elective subjects by inputing the subject codes (seperate by comma and space):ISIT315
NOTICE: Enrolment almost complete! (this is a finally clause)
Please select 3 more elective subjects by inputing the subject codes (seperate by comma and space):ISIT315

Already enrolled in subject.
Please enter only valid subject codes. NOTE: if multiple must be seperated by ', '
NOTICE: Enrolment almost complete! (this is a finally clause)
Please select 3 more elective subjects by inputing the subject codes (seperate by comma and space):CSCI318, ISI
T219, CSCI334
NOTICE: Enrolment almost complete! (this is a finally clause)

Congratulatoins. You had completed the enrolment to Bachelor of Computer Science.
--------------------
Student: John Smith (23454, male, 12/12/2000)
```



```
Cores:
CSIT111 (Programming Fundamentals, 6pt)
CSIT113 (Problem Solving, 6pt)
CSIT114 (System Analysis, 6pt)
CSIT115 (Data Management and Security, 6pt)
CSIT121 (Object Oriented Design and Programming, 6pt)
CSIT127 (Networks and Communications, 6pt)
CSIT128 (Introduction to Web Technology, 6pt)
CSCI235 (Database Systems, 6pt)
CSCI251 (Advanced Programming, 6pt)
CSIT214 (IT Project Management, 6pt)
MATH221 (Mathematics for Computer Science, 6pt)
CSCI203 (Algorithms and Data Structures, 6pt)
CSIT226 (Human Computer Interaction, 6pt)
CSIT314 (Software Development Methodologies, 6pt)
CSIT321 (Project, 12pt)

Big Data Major
CSCI317 (Database Performance Tuning, 6pt)
INFO411 (Data Mining and Knowledge Discovery, 6pt)
CSCI316 (Big Data Mining Techniques and Implementation, 6pt)
ISIT312 (Big Data Management, 6pt)
```



```
Electives:
ISIT315 (Semantic Web, 6pt)
CSCI318 (Software Engineering Practices & Principles, 6pt)
ISIT219 (Knowledge and Information Engineering, 6pt)
CSCI334 (Software Design, 6pt)
------------------
Total Enrolled Credit: 144pt
```