

CSL222-18 Object Oriented Programming

ASSIGNMENT 2

MARASIGAN, VEM AIENSI A.
2BSCS-1

OUTPUT

```
COMReportVersion2 <
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:D:\Programs\IntelliJ IDEA Community Edi
COM Report Version 2 {OOP Laboratory Assignment 2}
-Vem Aiensi A. Marasigan [2 BSCS 1]
=====

Block Section Code: 2BSIT-3
Description: Hacker Group
Adviser: T. Alcantara

Total Students: 3
-----
Student Number: 123
Student Name: John
Program: BSIT
Total Units enrolled: 5

Course Code      Description      Units  Day    Time
ITL221-18        OOP Lab              1      Mon    7:00 - 10:00AM
CIT221-18        OOP Lec              2      Tue    10:00 - 12:00PM
CIT223-18        Info Management 2    2      Mon    12:00 - 2:00PM

Student Number: 124
Student Name: Mary
Program: BSCS
Total Units enrolled: 4

Course Code      Description      Units  Day    Time
ITL221-18        OOP Lab              1      Mon    7:00 - 10:00AM
CIT221-18        OOP Lec              2      Tue    10:00 - 12:00PM
ITLEL2-18        IT Elective 2        1      Tue    1:00 - 4:00PM

Student Number: 125
Student Name: Ana
Program: BSEMC
Total Units enrolled: 6

Course Code      Description      Units  Day    Time
GECLWR-18        Life and Works Rizal  3      Mon    2:30- 5:30PM
CIT225-18        Quantitative Methods  3      Fri    1:00 - 4:00PM

=====

Block Section Code: 2BSCS-1
Description: Machine Learning Group
Adviser: J. Lima

Total Students: 4
-----
Student Number: 126
Student Name: Oliver
Program: BSCS
Total Units enrolled: 7

Course Code      Description      Units  Day    Time
CIT223-18        Info Management 2    2      Wed    8:00 - 10:00AM
ITL221-18        OOP Lab              1      Thu    7:00 - 10:00AM
CIT221-18        OOP Lec              2      Thu    4:00 - 6:00PM
ITL223-18        Info Management 2 Lab 2      Fri    7:00 - 10:00AM

Student Number: 127
Student Name: Nancy
Program: BSCS
Total Units enrolled: 3

Course Code      Description      Units  Day    Time
CIT223-18        Info Management 2    2      Wed    8:00 - 10:00AM
ITL221-18        OOP Lab              1      Sat    7:00 - 10:00AM

Student Number: 128
Student Name: Peter
Program: BSCS
Total Units enrolled: 5

Course Code      Description      Units  Day    Time
ITL221-18        OOP Lab              1      Thu    7:00 - 10:00AM
CIT221-18        OOP Lec              2      Thu    4:00 - 6:00PM
ITL223-18        Info Management 2 Lab 2      Fri    7:00 - 10:00AM

Student Number: 129
Student Name: Megan
Program: BSCS
Total Units enrolled: 3

Course Code      Description      Units  Day    Time
ITL221-18        OOP Lab              1      Thu    7:00 - 10:00AM
CIT221-18        OOP Lec              2      Thu    4:00 - 6:00PM

=====

Block Section Code: 2BSIT-2
Description: Network Security Group
Adviser: Y. Pasiol

Total Students: 2
-----
Student Number: 130
Student Name: Steve
Program: BSIT
Total Units enrolled: 7

Course Code      Description      Units  Day    Time
ITLEL2-18        IT Elective 2        1      Mon    1:00 - 4:00PM
GECLWR-18        Life and Works Rizal  3      Tue    2:30- 5:30PM
CIT225-18        Quantitative Methods  3      Wed    1:00 - 4:00PM

Student Number: 131
Student Name: Judel
Program: BSIT
Total Units enrolled: 12

Course Code      Description      Units  Day    Time
GECLWR-18        Life and Works Rizal  3      Tue    2:30- 5:30PM
CIT225-18        Quantitative Methods  3      Wed    1:00 - 4:00PM
ITL221-18        OOP Lab              1      Thu    7:00 - 10:00AM
CIT221-18        OOP Lec              2      Sat    10:00 - 12:00PM
ITLEL2-18        IT Elective 2        1      Sat    1:00 - 4:00PM
CIT223-18        Info Management 2    2      Sat    4:00 - 6:00PM

=====

Process finished with exit code 0
```

ASSIGNMENT 2**MARASIGAN, VEM AIENSI A.****2BSCS-1****SOURCE CODE -MAIN-**

```

1  package ui;
2
3  import dao.BlockSectionDA;
4
5  import java.io.FileNotFoundException;
6
7  no usages
8  public class COMReportVersion2 {
9      no usages
10     public static void main (String [] args) throws FileNotFoundException {
11         String report = "COM Report Version 2 {00P Laboratory Assignment 2}\n" +
12             "-Vem Aiens1 A. Marasigan [2 BSCS 1]\n" +
13             "=====\\n";
14
15         BlockSectionDA blockSection = new BlockSectionDA();
16         for(int count0 = 0; count0 < blockSection.getBlockList().size(); count0++)
17         {
18             report += "\\nBlock Section Code: " + blockSection.getBlockList().get(count0).getBlockCode()
19                 + "\\nDescription: " + blockSection.getBlockList().get(count0).getDescription()
20                 + "\\nAdviser: " + blockSection.getBlockList().get(count0).getAdviser()
21                 + "\\n\\n"
22                 + "Total Students: " + blockSection.getBlockList().get(count0).getTotalStudents()
23                 + "\\n-----\\n";
24             for(int count1 = 0; count1 < blockSection.getBlockList().get(count0).getTotalStudents(); count1++)
25             {
26                 report += "Student Number: " + blockSection.getBlockList().get(count0).getStudentList().get(count1).getStudentNumber()
27                     + "\\nStudent Name: " + blockSection.getBlockList().get(count0).getStudentList().get(count1).getName()
28                     + "\\nProgram: " + blockSection.getBlockList().get(count0).getStudentList().get(count1).getProgram()
29                     + "\\nTotal Units enrolled: " + blockSection.getBlockList().get(count0).getStudentList().get(count1).getTotalUnitsEnrolled()
30                     + "\\n\\n"
31                     + "Course Code\\t\\tDescription\\t\\t\\t\\tUnits\\tDay\\tTime\\n";
32
33                 for(int count2 = 0; count2 < blockSection.getBlockList().get(count0).getStudentList().get(count1).getCourseList().size(); count2++)
34                 {
35                     report += blockSection.getBlockList().get(count0).getStudentList().get(count1).getCourseList().get(count2).getCourseCode()
36                         + "\\t\\t" + wordAligner(blockSection.getBlockList().get(count0).getStudentList().get(count1).getCourseList().get(count2).getDescription())
37                         + "\\t\\t" + blockSection.getBlockList().get(count0).getStudentList().get(count1).getCourseList().get(count2).getUnit()
38                         + "\\t\\t" + blockSection.getBlockList().get(count0).getStudentList().get(count1).getCourseList().get(count2).getDay()
39                         + "\\t\\t" + blockSection.getBlockList().get(count0).getStudentList().get(count1).getCourseList().get(count2).getTime()
40                         + "\\n";
41                 }
42                 report += "\\n";
43             }
44             report += "\\n=====\\n";
45         }
46         System.out.println(report);
47     }
48     1 usage
49     static String wordAligner(String word)
50     { //this method transforms the word length to a fixed length for alignment purposes
51         String fixed = word;
52         while(fixed.length() < 20)
53             fixed += " ";
54         return fixed;
55     }
56 }

```

ASSIGNMENT 2

MARASIGAN, VEM AIENSI A.

2BSCS-1

SOURCE CODE -BlockSection & BlockSectionDA-

```

COMReportVersion2.java x BlockSection.java x BlockSectionDA.java x Course.java x COMReportVersion2.java x BlockSection.java x BlockSectionDA.java x Course.java x CourseDA.java x Student.java
1 package domain;
2
3 import java.util.List;
4
5 7 usages
6 public class BlockSection {
7     2 usages
8     private String blockCode;
9     2 usages
10    private String description;
11    2 usages
12    private String adviser;
13    2 usages
14    private Integer totalStudents;
15    2 usages
16    private List<Student> studentList;
17    1 usage
18    public String getBlockCode() {
19        return blockCode;
20    }
21
22    1 usage
23    public void setBlockCode(String blockCode) {
24        this.blockCode = blockCode;
25    }
26
27    1 usage
28    public String getDescription() {
29        return description;
30    }
31
32    1 usage
33    public void setDescription(String description) {
34        this.description = description;
35    }
36
37    1 usage
38    public String getAdviser() {
39        return adviser;
40    }
41
42    1 usage
43    public void setAdviser(String adviser) {
44        this.adviser = adviser;
45    }
46
47    2 usages
48    public Integer getTotalStudents() {
49        return totalStudents;
50    }
51
52    1 usage
53    public void setTotalStudents(Integer totalStudents) {
54        this.totalStudents = totalStudents;
55    }
56
57    10 usages
58    public List<Student> getStudentList() {
59        return studentList;
60    }
61
62    1 usage
63    public void setStudentList(List<Student> studentList) {
64        this.studentList = studentList;
65    }
66 }

```

```

1 package dao;
2
3 import domain.BlockSection;
4 import domain.Student;
5
6 import java.io.FileNotFoundException;
7 import java.io.FileReader;
8 import java.util.*;
9
10 3 usages
11 public class BlockSectionDA
12 {
13     4 usages
14     private List<BlockSection> blockList;
15
16     16 usages
17     public List<BlockSection> getBlockList() { return blockList; }
18
19     no usages
20     public void setBlockList(List<BlockSection> blockList) { this.blockList = blockList; }
21
22     1 usage
23     public BlockSectionDA() throws FileNotFoundException
24     {
25         Scanner blockInfo = new Scanner(new FileReader( fileName: "src/blockSection.csv"));
26         blockList = new ArrayList<BlockSection>();
27
28         while (blockInfo.hasNext())
29         {
30             String blockString = blockInfo.nextLine();
31             String [] blockData = new String [3];
32             blockData = blockString.split( regex: "," );
33
34             BlockSection block = new BlockSection();
35             StudentDA students = new StudentDA(blockData[0]);
36
37             block.setBlockCode(blockData[0]);
38             block.setDescription(blockData[1]);
39             block.setAdviser(blockData[2]);
40             block.setTotalStudents(students.getStudentList().size());
41             block.setStudentList(students.getStudentList());
42             blockList.add(block);
43         }
44     }

```

ASSIGNMENT 2

MARASIGAN, VEM AIENSI A.

2BSCS-1

SOURCE CODE -Student & StudentDA-

```
COMReportVersion2.java x BlockSection.java x BlockSectionDA.java x Course.java x Count x COMReportVersion2.java x BlockSection.java x BlockSectionDA.java x Course.java x CourseDA.java x Student.java
1 package domain;
2
3 import java.util.List;
4
5 11 usages
6 public class Student {
7     2 usages
8     private String studentNumber;
9     2 usages
10    private String name;
11    2 usages
12    private String program;
13    2 usages
14    private Integer totalUnitsEnrolled;
15    2 usages
16    private List<Course> courseList;
17
18    1 usage
19    public String getStudentNumber() {
20        return studentNumber;
21    }
22
23    1 usage
24    public void setStudentNumber(String studentNumber) {
25        this.studentNumber = studentNumber;
26    }
27
28    1 usage
29    public String getName() {
30        return name;
31    }
32
33    1 usage
34    public void setName(String name) {
35        this.name = name;
36    }
37
38    1 usage
39    public String getProgram() {
40        return program;
41    }
42
43    1 usage
44    public void setProgram(String program) {
45        this.program = program;
46    }
47
48    1 usage
49    public Integer getTotalUnitsEnrolled() {
50        return totalUnitsEnrolled;
51    }
52
53    1 usage
54    public void setTotalUnitsEnrolled(Integer totalUnitsEnrolled) {
55        this.totalUnitsEnrolled = totalUnitsEnrolled;
56    }
57
58    6 usages
59    public List<Course> getCourseList() {
60        return courseList;
61    }
62
63    1 usage
64    public void setCourseList(List<Course> courseList) {
65        this.courseList = courseList;
66    }
67 }

1 package dao;
2
3 import domain.Course;
4 import domain.Student;
5
6 import java.io.FileNotFoundException;
7 import java.io.FileReader;
8 import java.util.*;
9
10 2 usages
11 public class StudentDA
12 {
13     4 usages
14     private List<Student> studentList;
15
16     2 usages
17     public List<Student> getStudentList() { return studentList; }
18
19     no usages
20     public void setStudentList(List<Student> studentList) { this.studentList = studentList; }
21
22     1 usage
23     public StudentDA(String blockCode) throws FileNotFoundException
24     {
25         Scanner studentInfo = new Scanner (new FileReader( fileName: "src/studentInfo.csv"));
26         studentList = new ArrayList<Student>();
27
28         while (studentInfo.hasNext())
29         {
30             String studentString = studentInfo.nextLine();
31             String [] studentData = new String [4];
32             studentData = studentString.split( regex: "\\|");
33
34             if (blockCode.equals(studentData[0]))
35             {
36                 Student student = new Student();
37                 CourseDA courseDA = new CourseDA(studentData[1]);
38                 List<Course> courses= courseDA.getCourseList();
39
40                 //getting the total units the student enrolled based on courseList
41                 int totalUnits = 0;
42                 for(int count = 0; count < courses.size(); count++)
43                 {
44                     totalUnits += courses.get(count).getUnit();
45                 }
46
47                 student.setStudentNumber(studentData[1]);
48                 student.setName(studentData[2]);
49                 student.setProgram(studentData[3]);
50                 student.setTotalUnitsEnrolled(totalUnits);
51                 student.setCourseList(courses);
52
53                 studentList.add(student);
54             }
55         }
56     }
57 }
```


ASSIGNMENT 2

MARASIGAN, VEM AIENSI A.

2BSCS-1

SOURCE CODE -Course & CourseDA-

```

1 package domain;
2
3 public class Course {
4     private String courseCode;
5     private String description;
6     private Integer unit;
7     private String day;
8     private String time;
9
10    public String getCourseCode() {
11        return courseCode;
12    }
13
14    public void setCourseCode(String courseCode) {
15        this.courseCode = courseCode;
16    }
17
18    public String getDescription() {
19        return description;
20    }
21
22    public void setDescription(String description) {
23        this.description = description;
24    }
25
26    public Integer getUnit() {
27        return unit;
28    }
29
30    public void setUnit(Integer unit) {
31        this.unit = unit;
32    }
33
34    public String getDay() {
35        return day;
36    }
37
38    public void setDay(String day) {
39        this.day = day;
40    }
41
42    public String getTime() {
43        return time;
44    }
45
46    public void setTime(String time) {
47        this.time = time;
48    }
49 }

```

```

1 package dao;
2
3 import domain.Course;
4
5 import java.io.FileNotFoundException;
6 import java.io.FileReader;
7 import java.util.*;
8
9 public class CourseDA {
10
11    private List<Course> courseList;
12
13    public List<Course> getCourseList() { return courseList; }
14
15    public void setCourseList(List<Course> courseList) { this.courseList = courseList; }
16
17    public CourseDA(String studentNumber) throws FileNotFoundException {
18        Scanner schedInfo = new Scanner(new FileReader("src/scheduleInfo.csv"));
19        courseList = new ArrayList<Course>();
20
21        while (schedInfo.hasNext()) {
22            String courseString = schedInfo.nextLine();
23            String [] courseData = new String [6];
24            courseData = courseString.split(" ");
25
26            if (studentNumber.equals(courseData[0])) {
27                Course course = new Course();
28                course.setCourseCode(courseData[1]);
29                course.setDescription(courseData[2]);
30                course.setUnit(Integer.parseInt(courseData[3]));
31                course.setDay(courseData[4]);
32                course.setTime(courseData[5]);
33
34                courseList.add(course);
35            }
36        }
37    }
38 }

```