|  |
| --- |
|  |
| Project |
| SOFT6017 – Problem Solving & Programming II  (worth 20%) |
|  |
|  |
|  |

|  |
| --- |
|  |

Specification

The Goto Gym has recently started running evening classes for its members. The manager has decided to store the information about the classes in three text files: Instructors.txt, ClassDetails.txt and ClassTypes.txt.

Classes run from 7 to 10 on Mondays, Tuesdays, or Wednesdays. Classes are run on the hour at 7, 8, or 9 o'clock. All classes are exactly one hour in length. Three different types of classes are currently offered:

* Spinning
* Pilates
* Yoga

Each class type has a code made up of its first two letters in uppercase. For example, the code PI refers to the Pilates class. The code for the Yoga class is YO. The manager may decide to add more types of classes during the term, but plans on offering no more than 12 classes (of any type) in total. There can be multiple offerings of the same type of class. There are currently only four teachers/instructors, and no new instructors will be hired this term.

The information on when classes are currently running is stored in a text file called ClassDetails.txt. For each class that is timetabled, it stores:

* the class code,
* the day ( Mon, Tues, or Wed),
* the time,
* the instructor's first name.

Sample ClassDetails.txt:

PI

Mon

7

Amanda

PI

Mon

8

Amanda

YO

Wed

7

Gerry

SP

Tues

9

Frank

YO

Wed

9

Amanda

A separate file, ClassTypes.txt, contains the list of the current class types:

Pilates

Yoga

Spin

The last file, Instructors.txt, contains a list of names of the instructors who are currently available for this term. All instructors can teach all class types. No new instructors will be taken on this term. Sample content from the file is shown here:

Aidan

Amanda

Gerry

Frank

You are asked to develop the Goto Gym application. The application will need to load the information that is in the files into data structures so that the application can display and manipulate the data. You cannot assume that the information provided in the sample file contents above will match the information in the final text files, so do not hardcode the class types or the instructor names. When the application terminates, the contents of the ClassDetails.txt file must be updated to contain the updated class session information.

You should create a menu driven application that allows the manager to complete a series of tasks and then exit the application. The tasks MUST be listed in the same order as shown below:

The Goto Gym Classes

-------------------------

1. Add a New Class Session

2. Show Times of All Current Classes

3. Show Instructor Payments Due

4. Print Timetable for Instructor

5. Show Ordered Timetable with Codes

6. Exit

These tasks are described in more detail below, according to the menu option number. You may wish to develop them in a different order.

1. **Add a New Class Session**

Up to 12 class sessions may be added. The class type, the time, the day, and an instructor must be assigned for every new class session. Classes only run at 7, 8, or 9 o'clock from Monday to Wednesday. There is no limit to the number of classes that can run at a particular time.

**Advanced**: *Ideally the system should report instructor clashes and not assign an instructor to two classes at the same time.*

The Goto Gym Add a Class

-------------------------

Enter the class type:

1. Pilates

2. Yoga

3. Spin

1

Enter the day:

1. Mon

2. Tues

3. Wed

1

Choose a time:

1. 7 o'clock

2. 8 o'clock

3. 9 o'clock

1

Choose a teacher:

1. Amanda

2. Gerry

3. Aidan

4. Frank

3

Added PI on Mon at 7 with Aidan

1. **Show (display on the screen) All Current Class Sessions**

This lists all of the current class sessions that are running. The classes may appear in any order.

The Goto Gym Classes

-------------------------

PI Mon at 7 with Amanda

PI Mon at 8 with Amanda

YO Wed at 7 with Gerry

SP Tues at 9 with Frank

YO Wed at 9 with Amanda

**Advanced**: *List the classes ordered by the type of class*.  *If there is no class of a particular type running, the class is not listed. The following shows the current list of classes running:*

The Goto Gym Classes

-------------------------

Pilates

Mon at 7 with Amanda

Mon at 8 with Amanda

Yoga

Wed at 9 with Amanda

Wed at 7 with Gerry

Spin

Tues at 9 with Frank

1. **Show (display on the screen) Payments Due**

Display a list of payments due to instructors at the end of the week. For classes that run at 7 and 8, an instructor is given 60 Euros per hour. For 9 o'clock classes, the instructors are paid 80 Euros. If any instructor teaches more than 2 classes in a week, s/he is given 10 Euros extra for every class taught. The total wage bill for this week should also be displayed.

The Goto Gym Payments Due

-------------------------

Name Classes Pay (Euros)

---- ------- -----------

Amanda 3 230

Gerry 1 60

Aidan 0 0

Frank 1 80

Total wages for this week: 370.0 Euros

1. **Print An Instructor’s Timetable to a File**

Print a timetable of classes assigned to a particular instructor. This option creates a file that contains a list of the class sessions that have been assigned to him or her. The file is named after the instructor and contains the class type, the day, and the time. If there are no classes assigned to the instructor, a file is still created but contains the line: 'No classes assigned this week'. The information written to the file should also be echoed to the screen, as illustrated below:

The Goto Gym Timetables

-------------------------

Please enter a name: Amanda

AmandasTimetable.txt created.

Writing to file...

Pilates Mon at 7

Pilates Mon at 8

Yoga Wed at 9

The file AmandasTimetable.txt would be created in this example:

Pilates Mon at 7

Pilates Mon at 8

Yoga Wed at 9

1. **Show (display on the screen) a Timetable Ordered by Time**

**Advanced**: *This option shows a list of classes that are running in each particular time slot each day. Every possible time slot is listed. Beside each slot is a list of the codes of the classes running in that slot.*

Mon:

7 PI

8 PI

9

Tues:

7

8

9 SP

Wed:

7 YO

8

9 YO

Marking Scheme

|  |  |
| --- | --- |
| **Description** | **Mark** |
| **Code due Sunday April 30th 23:59**  Comments, e.g..:   * Appropriate and informative comments * Javadoc comments for every method   Programming Style, e.g.:   * Good program design, i.e. use of modular decomposition to structure solution and remove redundancy. * Naming Conventions for identifiers for classes, methods, variables and constants * Indentation and lineation * Readability of code | 10 |
| Implementation, e.g.:   * Creation of appropriate methods for menu options * Use of methods for particular tasks ie findTeacher * Use of arrays & parallel arrays * Use of files   Input Validation, e.g.:   * Checking that a value of the correct data type has been entered and that the value is in the appropriate range. * Ensuring the value entered corresponds to a valid menu option * Ensuring that file exists before opening etc.   Testing   * Testing of different scenarios for example: * Required files exist or files don’t exist * ClassDetails.txt contains 0 1 or more classes * Testing of payment for teacher with 1 2 or 3 classes * Testing of payment for teacher with classes before or after 9 * Enter the name of a teacher that does or doesn’t exist for timetable printing | 90 |
|  |  |