

Overview

To complete the week 7 practical, I had to first design and implement classes with attributes and methods that were suitable and moreover would allow me to create workshops, presentations, people and presenters. I then went on to create instances of the classes (objects) with various attribute values to represent specific instances of the classes. Finally, the program would use the objects and their methods to carry out a variety of different tasks.

Design

Each workshop has a date, venue, registration fee, 20 attendees and 10 presentations. Each presentation has a title, time, duration in minutes and the presenter. Each presenter has a name, affiliation and presentation fee. Finally, each attendee has a name, email address and an indication of whether they are a student.

The main class contains numerous methods, variables and constants. The main method manually calls the first three methods for every new object created. The first three methods are then used to create the following objects: Presentations, Presenters and People. The return types of the methods are objects and each take in parameters that are the attributes of the objects. I have decided to use multiple text files which contain a verity of names, affiliations, email addresses and presenter names. This allowed me to generate certain attributes of objects by reading in date from a text file. This was quicker than inputting every name manually and more over it meant if a user wishes to add, for example, a new name they can simply edit the text file rather than manually changing the code. So that the main class can track and manipulate the objects created, each object from the same class is stored in an array of that object. Moreover, this meant that I could cycle through the array handing every object from a specific class to a method in another class. This technique has allowed me to add the all the presentations objects and people objects, created to the workshop. Finally, the user can choose whether they wish to view all the presenters of the presentations and or all the attendees of the workshop. It should be noted that to add attendees to workshop a user will have to manually alter the code.

The first class I made was the workshop class. There was only one instance of this class which all the instances of the presentation and person classes could be added too. In addition to the necessary attributes for the workshop class I have added other attributes. These include workshop state which is used to indicate whether the workshop has made money, broken even or lost money. Following this I have used two integer variables which hold the total cost to put on the the workshop and the total amount of fees paid by attendees. All attendees that are students also receive a 30% discount, this is achieved by multiplying the total value of fees collected by attendees whom are students by 0.7. I have used two arrays of objects, these are used to keep track of the number of presentations and attendees in the workshop. There are numerous methods in the workshop class, the first method compares all the total costs of the workshop with the total paid by attendees, the method then returns a string which can be "Surplus", "Deficit" or "Broken even". The next two methods are used to add attendees and presentations to respective arrays, which allows the workshop object to manipulate the attributes of each object within the arrays. The workshop class also contains a method which allows the program to calculate the number of attendees that have signed up and thus tell whether or not the workshop is full. The final two methods which the workshop class can use are designed to check whether a valid date has been entered for the date of the workshop. The method takes in a string then checks its in the right format, if so it then converted into a date and then compared to the current date. If the date entered is in the future is is valid and the input string is then returned. However, if the date entered is in the past it deemed invalid and the user is prompted to re enter a valid date.

The next class is the presentation class. This class has the following attributes: title and time which are strings, cost which as an integer, duration which a float and the final attribute, presenter which is another object; presenter. So that one presenters can present multiple presentations but only ever one presentation can be presented by one presenter I have made a method which takes a presenter in as a parameter and then sets the presenter attribute, to presenter passed in. Then in the presenter class there is a presentation array attribute which can contain up to 10 presentations and so the same method then sets the fills the next available slot of the presentation array attribute of presenter handed in as a parameter, to the presentation used to call the method. The other method in the presentations class is used to print the details of a presentation. The duration attribute of this class is a randomly generated number between 1 and 4. This number then has the fraction 7/6 added to it, this is to allow a 5-minute overlap for each presentation, both at the start and at the end of the presentation when the duration is converted from ours to minutes.

A class called presentations is the next class, this class has the following attributes: name and affiliation which are both of type string and are chosen from a text file and presentation fee which is an integer that is randomly generated for each instance of the presenter class; as mentioned above each instance of the

presenter class can have up to 10 presentations. To make this possible I have created an array of tie Presentations with 10 spaces. There is then a method which fills the next free slot of the array with the passed in presentation. This provided a neat, many to one relationship between presenters and presentations. The final class is the person class, this class contains the following attributes name and email which are of type string and student which is Boolean. The student attribute is randomly set to true or false and based on whether it true or false, will the person receive discounts when they signup to a workshop.

I have also created an extension which allows objects to be automatically generated and then stored in an array. This meant that the user could choose how many people, presentations and workshops they wished to create. I did this because I recognised there was a serious limitation to my initial program. This limitation was that, if you wished to have a large amount of attendee's presentations, presenters and workshops, you would need an excessive amount of repeated code which is not efficient for many reasons. I achieved this by creating static arrays in each class which where used to store all the instances of that class. It was then a simple case of using a for loop to set the attributes of each object in the array.

Testing

To test my program and make sure it works as expected I will have to create a verity of objects from each class and then make sure each method from every class is called and works correctly. Because there are not many instances where the user needs to input data into the program, I will not have to do to many verification and validation tests, so for example entering a number instead of a string for a user name. Moreover, I have also incorporated many try and catch error statements which allow my program to catch many other errors. However, I will, where possible, enter invalid data in an attempt to break the program and therefore preform both extreme and erroneous tests. If the program works as expected, the program will be able to deal with the tests without crashing or returning inaccurate data. Testing the mathematical and logical operands in the program is also very important to ensure the data is being correctly manipulated.

Finally, although this wont effect the program itself, I will make sure the text output is formatted in way which is suitable for the user.

Examples

```
Please eneter a date for the workshop in the form dd/mm/yyyy :
HELLO
Invalid input!
20/09/2012
Date can not be in the past, please re-eneter a date
20/09/2016
Please eneter the workshop venue :
```

Here I tried to enter something that was in the format of a date, the program recognised this and prompted me to re-enter a valid date. I then tried to enter a date that was in the past, again the program recognised this and asked me re-enter a date that was not in the past. Finally, when I entered a date that was valid the program let me continue.

```
20/09/2016
Please eneter the workshop venue :
Jack Hall
Please eneter the workshop registration fee :
HELLO
Make sure you enter an integer
10
```

Again I couldn't enter an invalid registration fee

```
Error workshop already full, no more attendees can be added
Error no more presentations can be added to this workshop
Please eneter a date for the workshop in the form dd/mm/yyyy :
```

Here I added an extra person and an extra presentation, this was to try and make the program through an array index out of range exception and crash. However instead the program displayed the appropriate errors, then did not add the extra person and presentation and finally continued on as

```
#####
# WORKSHOP DETAILS #####
#####

Date : 20/09/2016, Venue: Jack Hall, Total cost of workshop : £303
Registration fees collected : £170.0 THEREFORE STATUS : DEFICIT

Presentation title: Steering (Initial membership)
Time: 19:00 - Duration (5 min overlap): 190.00 minutes

Presentation title: In Vivo -- In Silico
Time: 18:00 - Duration (5 min overlap): 190.00 minutes

Presentation title: Ubiquitous Computing: Experience, Design and Science
Time: 12:00 - Duration (5 min overlap): 70.00 minutes

Presentation title: Memories for Life
Time: 16:00 - Duration (5 min overlap): 190.00 minutes

Presentation title: Ubiquitous Computing: Experience, Design and Science
Time: 17:00 - Duration (5 min overlap): 130.00 minutes

Presentation title: The Architecture of Brain & Mind
Time: 12:00 - Duration (5 min overlap): 130.00 minutes

Presentation title: Dependable Systems Evolution
Time: 09:00 - Duration (5 min overlap): 250.00 minutes

Presentation title: Journeys in Nonclassical Computation
Time: 18:00 - Duration (5 min overlap): 250.00 minutes

Presentation title: Learning for Life
Time: 17:00 - Duration (5 min overlap): 130.00 minutes

Presentation title: Bringing the Past to Life for the Citizen
Time: 18:00 - Duration (5 min overlap): 130.00 minutes

THERE ARE CURRENTLY: 20 ATTENDEES SIGNED UP -- WORKSHOP FULL
```

Here you can see that the program has generated a workshop with the correct date, venue and also calculated the total cost of the workshop. Moreover, the program has then calculated the total fees collected and then compared it to the cost of the workshop. Because the workshop cost more than the fees collected, deficit is printed. Finally, the program has correctly displayed the presentations with the correct durations that include a 5-minute overlap.

Here you can see that there are 20 attendees and thus the above screen shot which says the workshop is full was correct.

Moreover, there are 10 attendees whom are students and 10 attendees whom are not students. Because the sign up fee was £10 the over fees collected should be $10*10 + 10*(10*7)$. This equals 170 therefore the program was correct.

You can also see that the sum of all the presenter fees is 303 so therefore again the program correctly calculated this.

Finally, the program has correctly displayed deficit as the presenter fees exceed the fees paid by attendees.

```
THERE ARE CURRENTLY: 20 ATTENDEES SIGNED UP -- WORKSHOP FULL

Please enter a choice:
1. Display attendees
2. Display presenters
3. Quit
1
Student: true Name: Tamekia Pardee Email: stone@meekness.com
Student: false Name: Dianna Wallner Email: ca-tech@dps.centrin.net.id
Student: false Name: Nichol Lahman Email: trinanda_lestyowati@telkomsel.co.id
Student: true Name: Retta Deblois Email: asst_dos@astonrasuna.com
Student: true Name: Mac Nakayama Email: amartabali@dps.centrin.net.id
Student: true Name: Kimberly Sgro Email: achat@cbn.net.id
Student: false Name: Lavonia Riker Email: bali@tuguhotels.com
Student: true Name: Stephan Outen Email: baliminimalist@yahoo.com
Student: true Name: Karrie Shamburger Email: bliss@thebale.com
Student: false Name: Kattie Frances Email: adhidharma@denpasar.wasantara.net.id
Student: false Name: Cassie Bushnell Email: centralreservation@amayanahotel.com
Student: false Name: Janette Rameau Email: apribadi@balimandira.com
Student: true Name: Shawanna Favreau Email: cdagenhart@ifc.org
Student: false Name: Mariette Clutter Email: dana_supriyanto@interconti.com
Student: false Name: April Sitler Email: dos@novotelbali.com
Student: true Name: Desmond Bacote Email: daniel@hotelpadma.com
Student: false Name: Mao Corbin Email: daniel@balibless.com
Student: true Name: Dorothea Houchin Email: djoko_pe@jayakartahotelsresorts.com
Student: true Name: Gisela Brumley Email: expdepot@indosat.net.id
Student: false Name: Luanne Bakke Email: feby.adamsyah@idn.xerox.com
2
Name : Zack Ferriera , Affiliation : Accounting & Finance Presenter Fee: £26.00
Name : Katharina Lakin , Affiliation : Aeronautical & Manufacturing Engineering Presenter Fee: £16.00
Name : Sparkle West , Affiliation : Agriculture & Forestry Presenter Fee: £49.00
Name : Shalonda Taveras , Affiliation : American Studies Presenter Fee: £14.00
Name : Lanora Schuller , Affiliation : Anatomy & Physiology Presenter Fee: £32.00
Name : Breanna Belew , Affiliation : Anthropology Presenter Fee: £44.00
Name : Gaston Lash , Affiliation : Architecture Presenter Fee: £34.00
Name : Brenna Sartwell , Affiliation : Art & Design Presenter Fee: £29.00
Name : Margaretta Sabella , Affiliation : Biological Sciences Presenter Fee: £32.00
Name : Davida Castrejon , Affiliation : Business & Management Studies Presenter Fee: £27.00
3
8afbf723:W07-Practical henryhargreaves$
```

```
8afb723:W07-Practical henryhargreaves$ java W07Practical
Please enter a date for the workshop in the form dd/mm/yyyy :
20/09/2016
Please enter the workshop venue :
Jack Hall
Please enter the workshop registration fee :
60

##### WORKSHOP DETAILS #####
Date : 20/09/2016, Venue: Jack Hall, Total cost of workshop : £265
Registration fees collected : £966.0 THEREFORE STATUS : SURPLUS
```

Here when I entered a larger registration fee the program then out surplus

I removed a person from the code and ran it again now the program correctly shows that there are only 19 attendees signed up and that the workshop is not full

THERE ARE CURRENTLY: 19 ATTENDEES SIGNED UP

```
Please enter a choice:
1. Display attendees
2. Display presenters
3. Quit
1
Student: false Name: Tamekia Pardee Email: stone@meekness.com
Student: false Name: Dianna Wallner Email: ca-tech@dps.centrin.net.id
Student: false Name: Nichol Lahman Email: trinanda_lestyowati@telkomsel.co.id
Student: true Name: Retta Deblis Email: asst_dos@astonrasuna.com
Student: true Name: Mac Nakayama Email: amartabali@dps.centrin.net.id
Student: true Name: Kimberly Sgro Email: achatv@cbn.net.id
Student: true Name: Lavonia Riker Email: bali@tuguhotels.com
Student: false Name: Stephan Outen Email: baliminimalist@yahoo.com
Student: true Name: Karrie Shamburger Email: bliss@thebale.com
Student: true Name: Kattie Frances Email: adhidharma@denpasar.wasantara.net.id
Student: false Name: Cassie Bushnell Email: centralreservation@ramayanahotel.com
Student: false Name: Janette Rameau Email: apribadi@balimandira.com
Student: false Name: Shawanna Favreau Email: cdagenhart@ifc.org
Student: false Name: Mariette Clutter Email: dana_supriyanto@interconti.com
Student: false Name: April Sitler Email: dos@novotelbali.com
Student: true Name: Desmond Bacote Email: daniel@hotelpadma.com
Student: false Name: Mao Corbin Email: daniel@balibless.com
Student: false Name: Dorothea Houchin Email: djoko_p@jayakartahotelsresorts.com
Student: true Name: Gisela Brumley Email: expdepot@indosat.net.id
```

```
8afb723:W07PracticalCopy henryhargreaves$ java W07Practical
Please enter the number of people to create
20
Please enter the number of presentations to create
20
Please enter the number of presenters to create
20
Please enter the workshop date:
20/09/2016
1. ##### WORK SHOP 1 date : 20/09/2016, venue: Jack Hall #####
Registration fees :0
name : Zack Ferriera affiliation : Accounting & Finance cost: £13
name : Katharina Lakin affiliation : Aeronautical & Manufacturing Engineering cost: £26
name : Sparkle West affiliation : Agriculture & Forestry cost: £30
name : Shalonda Taveras affiliation : American Studies cost: £39
name : Lanora Schuller affiliation : Anatomy & Physiology cost: £26
name : Breonna Below affiliation : Anthropology cost: £19
name : Gaston Lash affiliation : Architecture cost: £14
name : Brenna Sartwell affiliation : Art & Design cost: £39
name : Margareta Sabella affiliation : Biological Sciences cost: £30
name : Davida Castrejon affiliation : Business & Management Studies cost: £43
name : Nobuko Kreiger affiliation : Chemical Engineering cost: £43
name : Iona Pinzon affiliation : Chemistry cost: £44
name : Kyra Mitchum affiliation : Civil Engineering cost: £42
name : Shonta Byard affiliation : Classics & Ancient History cost: £28
name : Damaris Culbert affiliation : Communication & Media Studies cost: £40
name : Roland Rommel affiliation : Complementary Medicine cost: £30
name : Jennette Mallon affiliation : Computer Science cost: £37
name : Sondra Saia affiliation : Counselling cost: £31
name : Debra Flanery affiliation : Criminology cost: £44
name : Kami Newcombe affiliation : Dentistry cost: £21
```

Here you can see that when I executed my extension and entered the number of attendees, presentations and presenters again the program executed correctly

Evaluation

My program successfully meets all the initial requirements for the program and moreover I have also added extensions in. The core functionality of my program successfully works and is also able to with erroneous and extreme tests without crashing but instead recognising the errors. The user had manually change the code to add and remove attendees, presenters, presentations and workshops which I could lead to confusion and accidental alteration of the code. However, the integration of using text files to store data used for the attributes of the objects meant was a good compromise. Moreover, the extension was able to randomly generate multiple objects and thus meaning the source code never needed to be altered.

Conclusion

In conclusion I'm happy with end program(s). They complete the task set and have a verity of added functionality which make the program more flexible and user friendly. The extension was very hard to program and does contain some bugs which are exposed when carrying out extreme and erroneous testing. If I had more time I would go and fix. One other thing I may change is where the methods are, for example the workshop class contains a large quantity of methods and the person class only contains one method. Although the overall functionality of the program is not effected by this, if I wished to use the classes I created in other programs I may have to re-write methods. Again due to limitations of command line I would have rather made and interactive graphical user interface but this was not possible.