REVISION FOR FINAL EXAM (SECJ1023)
PROGRAMMING TECHNIQUE II
SECTION 04 & 05, SEM 2, 2022/2023

QUESTION 1

Faculty of Computing will be organizing a competition that is open to all UTM students. The competition is meant to provide a platform to the students to exhibit their idea. Some important notes regarding the competition:

a) It is an individual participation.

b) Each student must have a project beforehand.

c) Each student must find an advisor which can be any UTM staff regardless of their position. That means, besides lecturers, technicians, research officers, etc, also can be appointed as advisers. However, only one advisor is allowed for each student.

d) Also, each staff can only advise a student.

Suppose the class design has been done by other parties as shown in Figure 1. Based on the class diagram in Figure 1 and the requirements above, write a C++ program that does the following tasks:

a) Implement all the classes and their relationships. You need to define all the methods listed in the diagram, and you can add other methods whenever possible.

b) Create a list of **`Advisor`** objects and another list of **`Students`** objects. Use array of objects and the sample data given in an input files.

c) Assign each student with an advisor.

d) Print the information of all the students along with their projects and advisers.

e) Suppose the first and second students want to change their advisor each other, while the advisor for the third student wants to withdraw from being appointed as an advisor. Write the code to implement these situations.

f) Print again the final list.

g) The output of the program is shown in **Figure 2**.

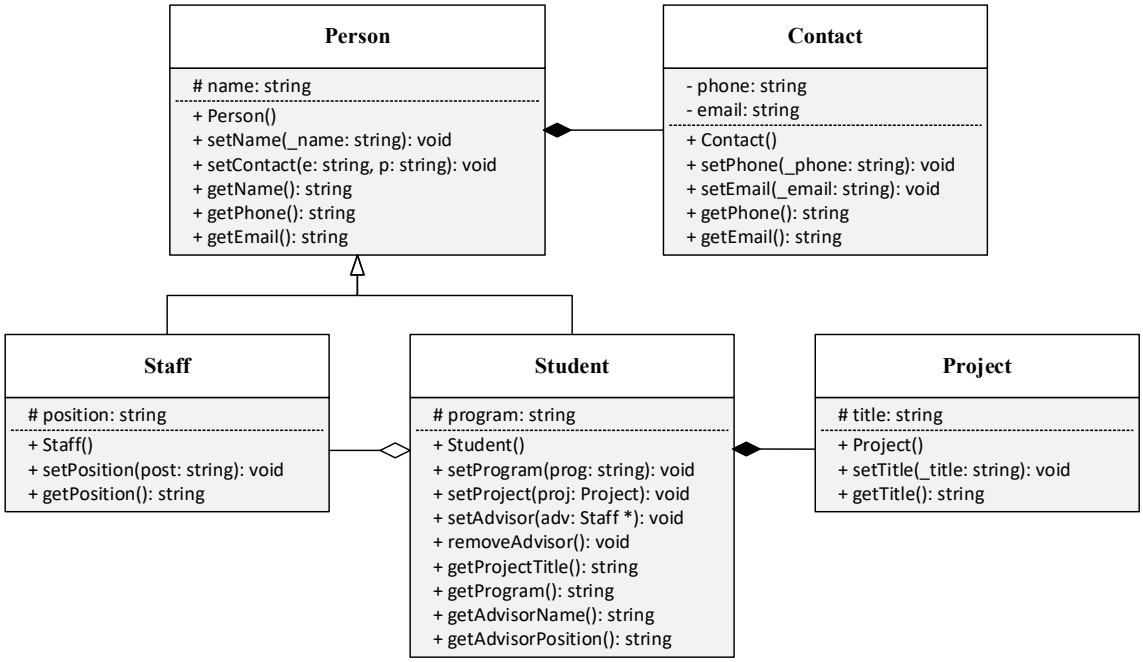

Figure 1: UML class diagram

Table 1: Sample data of advisors

No	Advisor's Name	Phone	Email	Position
1	Prof. Dr. Muhammad Roslan	0199875678	mroslana29@gmail.com	Lecturer
2	Mr. Qamarool Zaman	01156781234	qam23@gmail.com	Senior Technician
3	Dr. Siti Zubaidah Rosli	0101117456	ctzr983@gmail.com	Research Officer
4	AP. Dr. Karim Hisham	0123451222	kh123@gmail.com	Lecturer

Table 2: Sample data of students

No	Student's Name	Phone	Email	Program	Project
1	Lim Sew Ying	010897634045	lsying12@live.utm.my	Electrical Engineering	Smart Dustbin
2	Abu Bakar Razali	0111210000	abraz78@live.utm.my	Computer Science	Money Recognition for Blind People
3	Nur Amalina Muhammad	0129000123	nam978@live.utm.my	Biomedical Engineering	Dengue Testing Kit
4	Abas Amirullah	0121236782	abasa12@live.utm.my	Science	Sports in Education

Sample data of advisors and students are provided in the input files named **"inpAdvisor.txt"** and **"inpStudent.txt"**.

```
NO  NAME                    PHONE          EMAIL                 PROGRAM                 PROJECT                              ADVISOR
1   Lim Sew Ying            010897634045   lsying12@live.utm.my  Electrical Engineering  Smart Dustbin                        Prof. Dr. Muhammad Roslan
2   Abu Bakar Razali        0111210000     abraz78@live.utm.my   Computer Science        Money Recognition for Blind People   Mr. Qamarool Zaman
3   Nur Amalina Muhammad    0129000123     nam978@live.utm.my    Biomedical Engineering  Dengue Testing Kit                   Dr. Siti Zubaidah Rosli
4   Abas Amirullah          0121236782     abasa12@live.utm.my   Science                 Sports in Education                  AP. Dr. Karim Hisham


NO  NAME                    PHONE          EMAIL                 PROGRAM                 PROJECT                              ADVISOR
1   Lim Sew Ying            010897634045   lsying12@live.utm.my  Electrical Engineering  Smart Dustbin                        Mr. Qamarool Zaman
2   Abu Bakar Razali        0111210000     abraz78@live.utm.my   Computer Science        Money Recognition for Blind People   Prof. Dr. Muhammad Roslan
3   Nur Amalina Muhammad    0129000123     nam978@live.utm.my    Biomedical Engineering  Dengue Testing Kit
4   Abas Amirullah          0121236782     abasa12@live.utm.my   Science                 Sports in Education                  AP. Dr. Karim Hisham

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Figure 2: Output of the program

Question 2

Write a complete C++ program based on the UML class diagram given in **Figure 3**. Your program should be able to produce the output shown in **Figure 4**.

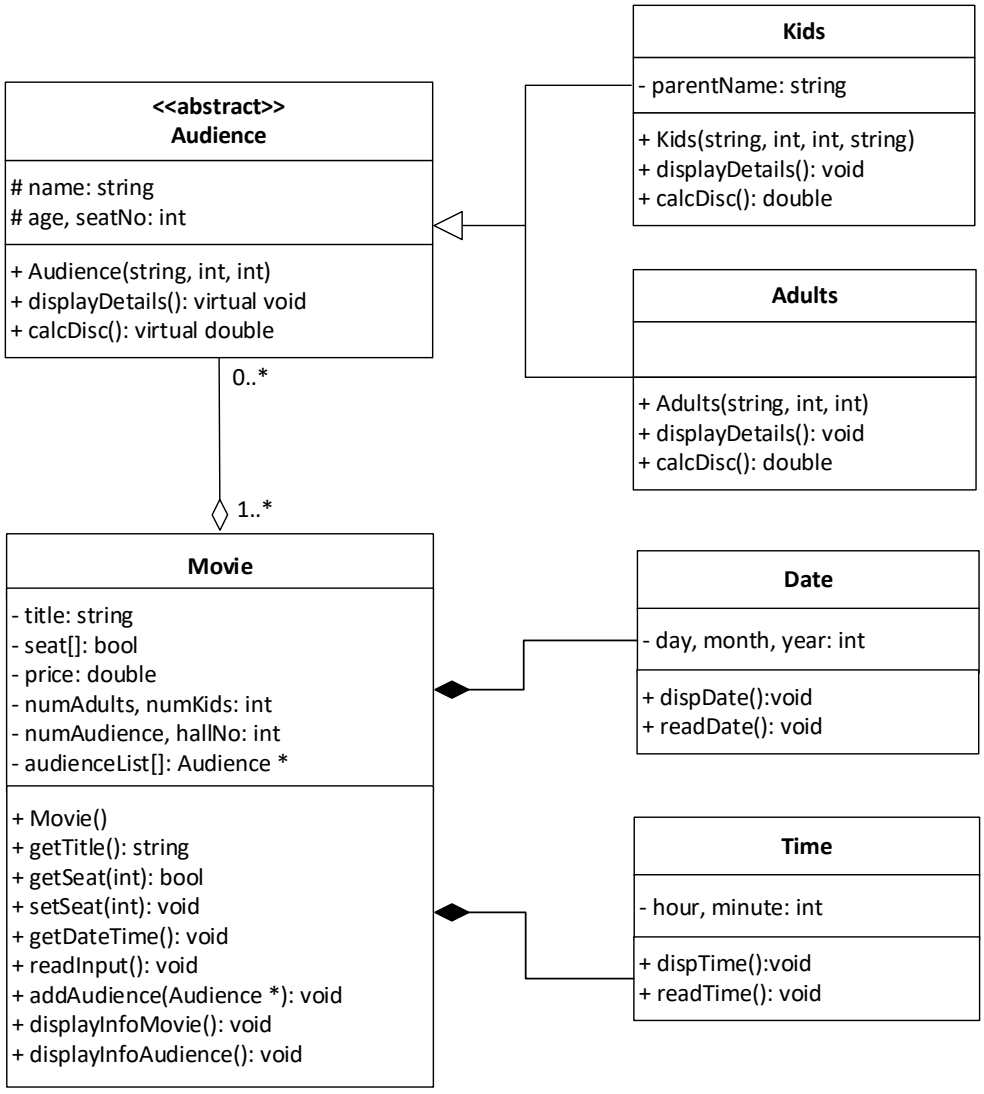

Figure 3: UML class diagram

Implement all the classes with the member data (attributes) and member functions (methods) specified in the diagram. The purpose of each function is as the name implies and some of them are further explained below. Write the program based on the following tasks:

Task 1:

In **`Audience`** class, define two more member functions and allow them to be polymorphic. The functions and their description are as follows:

a) **`displayDetails`**: to display an audience's name and age onto the screen in the following format (please refer to the sample output given in **Figure 4**):

```
????? ?????    ##
```

b) **`calcDisc`**: is a pure **`virtual`** function.

Task 2:

a) Specify the **`Adults`** class as a child of the **`Audience`** class.

The **`Adults`** class has three member functions:

b) Constructor with arguments: to initialize all the parent's attributes.

c) **`calcDisc`**: to return 20% discount received by a senior citizen. The age of senior citizen is 60 years or higher.

d) **`displayDetails`**: to display an audience's name, age, hyphen symbol (-) and seat number onto the screen in the following format (please refer to the sample output given in **Figure 4**):

```
????? ?????   ##     -            #
```

It needs to invoke the parent's **`displayDetails`** function.

Task 3:

a) Specify the **`Kids`** class as a child of the **`Audience`** class.

The **`Kids`** class has three member functions:

b) Constructor with arguments: to initialize all the member attributes for the class, including the parent's attributes.

c) **`calcDisc`**: to return 100% discount received by an infant. The age of infant is 2 years or less. It is also return 20% discount received by a kid. The kid is the audience 12 years of age or less.

d) **`displayDetails`**: to display an audience's name, age, parent name and seat number onto the screen. in the following format (please refer to the sample output given in **Figure 4**):

```
????? ?????   ##     ????? ?????  #
```

It needs to invoke the parent's **`displayDetails`** function.

Task 4:

In **`Movie`** class, do the following tasks:

a) Define all the member variables of the class.

b) Complete the **`readInput`** function definition. The function reads the values from the keyboard for title, hall number, ticket price, date and time of the movie. It needs to invoke the **`readDate`** and **`readTime`** functions.

c) Complete the **`addAudience`** function definition. The function assigns the element in the array of **`Audience`** pointers with the passed argument. It also updates the number of adults, kids and all audiences.

d) Complete the **`displayInfoMovie`** function definition. The function displays the title, hall number, date, time and ticket price of the movie in the following format (please refer to the sample output given in **Figure 4**):

```
Title: ????? ?????
Hall : #
Date : ##-##-####
Time : ##:## ??
Price: RM##.##
```

e) Complete the **`displayInfoAudience`** function definition. The function displays the information of all the audiences in the following format (please refer to the sample output given in **Figure 4**):

```
1.  ????? ?????   ##     -            #        ##.##
2.  ????? ?????   ##     -            #        ##.##
3.  ????? ?????   ##     ????? ?????  #        ##.##
4.    :     :      :       :          :        :  :
```

Task 5:

In **`main`** function, do the following tasks:

a) Enter the task chosen.

b) In **`case 1`**:
- Enter the details of movie.
- Add the **`Movie`** object into the **`Movie`** array.

c) In **`case 2`**:
- Display the list of movies' title along with its date and time.
- Check the seat availability for the audience more than 2 years old. If the seat is available, update the status of the seat.
- Dynamically allocates a new audience (an adult or a kid) object. The kid is the audience 12 years of age or less. ***Hint:*** Use a polymorphic concept.
- Add the audience to the selected **`Movie`** object.

d) In **`case 3`**, display the list of movies' details.

e) In **`case 4`**, display the list of audiences' details based on the movie.

f) Enter the task chosen.

Figure 4 shows a sample screen output that your program should produce. Note that, all the interactions shown in **Figure 4** are continuous in a single run. Note also that, the **bold** texts indicate input entered by the user.

```
========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audiences
[5] Exit
==========================

Select task: 3
Sorry!! No movie data to display...

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audiences
[5] Exit
==========================

Select task: 4
Sorry!! No movie, please add movie first...

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audiences
[5] Exit
==========================

Select task: 1

<<< Add Movie >>>
```

```
Title: Jumanji 2
Hall : 3
Ticket Price: RM28
Date [dd-mm-yyyy]: 30-5-2020
Time [hour:min] using 24-hour format: 11:30

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audiences
[5] Exit
==========================

Select task: 2

<<< Add Audience >>>

Movie list
1) Jumanji 2 (30-05-2020, 11:30 am)

Select movie: 1

--- Enter Audience Info ---
Name   : Afiqah Salim
Age    : 28
Seat No: 4

Press 'Y' to continue >> y

--- Enter Audience Info ---
Name   : Auni Nazri
Age    : 2
Parent Name: Afiqah Salim

Press 'Y' to continue >> n

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audiences
[5] Exit
==========================

Select task: 4

<<< Movie(s) and Audience(s) Info >>>

Number of Movies: 1

Movie #1
Title: Jumanji 2
Hall : 3
Date : 30-05-2020
Time : 11:30 am
Price: RM28.00

Number of Audience: 2
Number of Adults  : 1
Number of Kids    : 1

No  Name                    Age    Parent Name             Seat No  Ticket (RM)
1.  Afiqah Salim            28     -                       4        28.00
2.  Auni Nazri              2      Afiqah Salim            4        0.00

Total ticket price = RM28.00

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
```

```
[4] Display Audiences
[5] Exit
==========================

Select task: 1

<<< Add Movie >>>
Title: Frozen 2
Hall : 4
Ticket Price: RM24
Date [dd-mm-yyyy]: 30-5-2020
Time [hour:min] using 24-hour format: 16:05

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audiences
[5] Exit
==========================

Select task: 1

<<< Add Movie >>>
Title: Jumanji 2
Hall : 5
Ticket Price: RM30
Date [dd-mm-yyyy]: 31-5-2020
Time [hour:min] using 24-hour format: 22:15

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audiences
[5] Exit
==========================

Select task: 3

<<< Movie Info >>>

Number of Movies: 3

Movie #1
Title: Jumanji 2
Hall : 3
Date : 30-05-2020
Time : 11:30 am
Price: RM28.00

Movie #2
Title: Frozen 2
Hall : 4
Date : 30-05-2020
Time : 04:05 pm
Price: RM24.00

Movie #3
Title: Jumanji 2
Hall : 5
Date : 31-05-2020
Time : 10:15 pm
Price: RM30.00

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audiences
[5] Exit
==========================
```

```
Select task: 2

<<< Add Audience >>>

Movie list
1) Jumanji 2 (30-05-2020, 11:30 am)
2) Frozen 2 (30-05-2020, 04:05 am)
3) Jumanji 2 (31-05-2020, 10:15 am)

Select movie: 3

--- Enter Audience Info ---
Name   : Arif Irfan
Age    : 15
Seat No: 6

Press 'Y' to continue >> y

--- Enter Audience Info ---
Name   : Hakim Irfan
Age    : 20
Seat No: 6
Seat unavailable, choose others!

Seat No: 7

Press 'Y' to continue >> y

--- Enter Audience Info ---
Name   : Irfan Hanif
Age    : 60
Seat No: 8

Press 'Y' to continue >> y

--- Enter Audience Info ---
Name   : Haikal Irfan
Age    : 2
Parent Name: Irfan Hanif

Press 'Y' to continue >> n

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audiences
[5] Exit
==========================

Select task: 2

<<< Add Audience >>>

Movie list
1) Jumanji 2 (30-05-2020, 11:30 am)
2) Frozen 2 (30-05-2020, 04:05 am)
3) Jumanji 2 (31-05-2020, 10:15 am)

Select movie: 1

--- Enter Audience Info ---
Name   : Lisa Amin
Age    : 24
Seat No: 5

Press 'Y' to continue >> y

--- Enter Audience Info ---
Name   : Hanif Ayob
```

```
Age    : 26
Seat No: 4
Seat unavailable, choose others!

Seat No: 6

Press 'Y' to continue >> y

--- Enter Audience Info ---
Name   : Latif Hanif
Age    : 1
Parent Name: Hanif Ayob

Press 'Y' to continue >> n

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audience
[5] Exit
==========================

Select task: 4

<<< Movie(s) and Audience(s) Info >>>

Number of Movies: 3

Movie #1
Title: Jumanji 2
Hall : 3
Date : 30-05-2020
Time : 11:30 am
Price: RM28.00

Number of Audience: 5
Number of Adults  : 3
Number of Kids    : 2

No  Name                    Age    Parent Name             Seat No  Ticket (RM)
1.  Afiqah Salim            28     -                       4        28.00
2.  Auni Nazri              2      Afiqah Salim            4        0.00
3.  Lisa Amin               24     -                       5        28.00
4.  Hanif Ayob              26     -                       6        28.00
5.  Latif Hanif             1      Hanif Ayob              6        0.00

Total ticket price = RM84.00

Movie #2
Title: Frozen 2
Hall : 4
Date : 30-05-2020
Time : 04:05 am
Price: RM24.00

No audience!

Movie #3
Title: Jumanji 2
Hall : 5
Date : 31-05-2020
Time : 10:15 am
Price: RM30.00

Number of Audience: 4
Number of Adults  : 3
Number of Kids    : 1

No  Name                    Age    Parent Name             Seat No  Ticket (RM)
1.  Arif Irfan              15     -                       6        30.00
2.  Hakim Irfan             20     -                       7        30.00
3.  Irfan Hanif             60     -                       8        24.00
4.  Haikal Irfan            2      Irfan Hanif             8        0.00
```

```
Total ticket price = RM84.00

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audience
[5] Exit
==========================

Select task: 5
Thank you! :)
```

(a) Example output of the first run

```
========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audience
[5] Exit
==========================

Select task: 1

<<< Add Movie >>>
Title: Jumanji
Hall : 3
Ticket Price: RM28
Date [dd-mm-yyyy]: 30-5-2020
Time [hour:min] using 24-hour format: 16:05

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audience
[5] Exit
==========================

Select task: 2

<<< Add Audience >>>

Movie list
1) Jumanji (30-05-2020, 04:05 pm)

Select movie: 1

--- Enter Audience Info ---
Name   : Abu Ali
Age    : 25
Seat No: 6

Press 'Y' to continue >> y

--- Enter Audience Info ---
Name   : Ani Ali
Age    : 23
Seat No: 5

Press 'Y' to continue >> y

--- Enter Audience Info ---
Name   : Aydin Abu
Age    : 2
Parent Name: Abu Ali

Press 'Y' to continue >> n
```

```
========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audience
[5] Exit
==========================

Select task: 3

<<< Movie Info >>>

Number of Movies: 1

Movie #1
Title: Jumanji
Hall : 3
Date : 30-05-2020
Time : 04:05 am
Price: RM28.00

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audience
[5] Exit
==========================

Select task: 4

<<< Movie(s) and Audience Info >>>

Number of Movies: 1

Movie #1
Title: Jumanji
Hall : 3
Date : 30-05-2020
Time : 04:05 am
Price: RM28.00

Number of Audience: 3
Number of Adults  : 2
Number of Kids    : 1

No  Name                    Age    Parent Name             Seat No  Ticket (RM)
1.  Abu Ali                 25     -                       6        28.00
2.  Ani Ali                 23     -                       5        28.00
3.  Aydin Abu               2      Abu Ali                 5        0.00

Total ticket price = RM56.00

========== Menu ==========
[1] Add Movie
[2] Add Audience
[3] Display Movies
[4] Display Audience
[5] Exit
==========================

Select task: 5
Thank you! :)
```

(b) Example output of the second run

Figure 4: Example runs of program

Question 3

Write a complete C++ program based on the UML class diagram given in **Figure 5**. Your program should be able to produce the output shown in **Figure 6**.

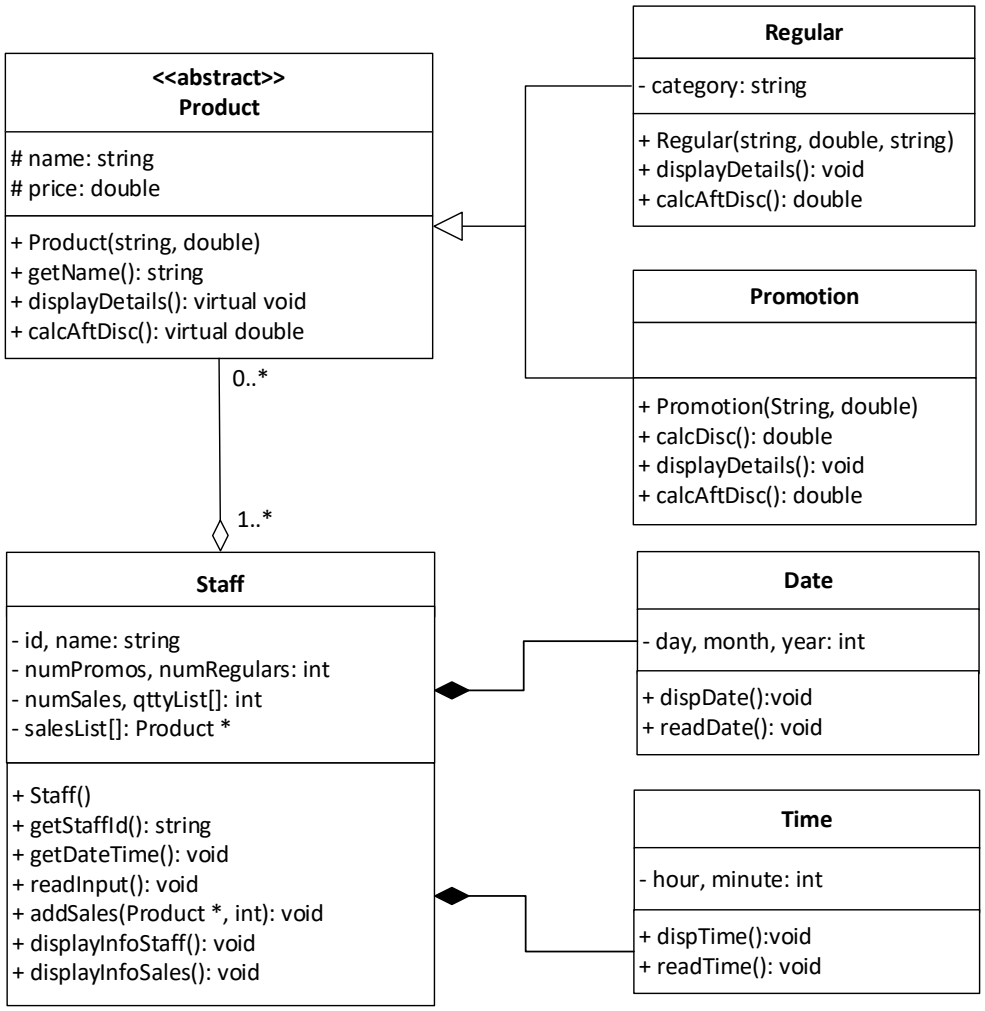

Figure 5: UML class diagram

Implement all the classes with the member data (attributes) and member functions (methods) specified in the diagram. The purpose of each function is as the name implies and some of them are further explained below. Write the program based on the following tasks:

Task 1:

In **`Product`** class, define two more member functions and allow them to be polymorphic. The functions and their description are as follows:

a) **`displayDetails`**: to display a product's name and price onto the screen in the following format (please refer to the sample output given in **Figure 6**):

```
????? ?????    ##.##
```

b) **`calcAftDisc`**: is a pure **`virtual`** function.

Task 2:

a) Specify the **`Promotion`** class as a child of the **`Product`** class.

The **`Promotion`** class has three member functions:

b) Constructor with arguments: to initialize all the parent's attributes.

c) **`calcDisc`**: to return discounted price. The discount rate of promotion product is 15%.

d) **`calcAftDisc`**: to return price after discount. It needs to invoke the **`calcDisc`** function.

e) **`displayDetails`**: to display a product's name, price and discounted price onto the screen in the following format (please refer to the sample output given in **Figure 6**):

```
????? ?????   ##.##     #.##
```

It needs to invoke the parent's **`displayDetails`** function.

Task 3:

a) Specify the **`Regular`** class as a child of the **`Product`** class.

The **`Regular`** class has three member functions:

b) Constructor with arguments: to initialize all the member attributes for the class, including the parent's attributes.

c) **`calcAftDisc`**: to return the price.

d) **`displayDetails`**: to display a product's name, price and whitespace onto the screen. in the following format (please refer to the sample output given in **Figure 6**):

```
????? ?????   ##.##     (whitespace)
```

It needs to invoke the parent's **`displayDetails`** function.

Task 4:

In **`Staff`** class, do the following tasks:

a) Define all the member variables of the class.

b) Complete the **`readInput`** function definition. The function reads the values from the keyboard for id, name, date and time check in of the staff. It needs to invoke the **`readDate`** and **`readTime`** functions.

c) Complete the **`addSales`** function definition. The function assigns the element in the array of **`Product`** pointers and the element in the integers array with the passed arguments. It also updates the number of promotion, regular and all sales.

d) Complete the **`displayInfoStaff`** function definition. The function displays the id, name, date, and time check in of the staff in the following format (please refer to the sample output given in **Figure 6**):

```
Id   : ?????
Name : ????? ?????
Date : ##-##-####
Time : ##:## ??
```

e) Complete the **`displayInfoSales`** function definition. The function displays the information of all the sales in the following format (please refer to the sample output given in **Figure 6**):

```
1.  ????? ?????   ##.##    #.##     #        ##.##
2.  ????? ?????   ##.##    #.##     #        ##.##
3.  ????? ?????   ##.##              #        ##.##
4.    :     :      :  :      :      :        :  :
```

Task 5:

In **`main`** function, do the following tasks:

a) Enter the task chosen.

b) In **`case 1`**:
- Enter the details of staff.
- Add the **`Staff`** object into the **`Staff`** array.

c) In **`case 2`**:
- Enter the details of product: the type, name, and price of the product.
- For the regular product, enter the category of the product and then update the name of the product by appending it with category (please refer to the sample output given in **Figure 6**).
- Dynamically allocates a new product (a promotion or regular) object based on the product type. ***Hint:*** Use a polymorphic concept.
- Add the **`Product`** pointer into the **`Product`** pointers array.

d) In **`case 3`**:
- Display the list of staffs' id.
- Display the list of products' name.
- Add sales of product to the selected **`Staff`** object.

e) In **`case 4`**, display the list of sales' details based on the staff.

f) Enter the task chosen.

Figure 6 shows a sample screen output that your program should produce. Note that, all the interactions shown in **Figure 6** are continuous in a single run. Note also that, the **bold** texts indicate input entered by the user.

```
========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 3
Sorry!! No staff or product, please add staff or product first...

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 4
Sorry!! No staff, please add staff first...

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 1

<<< Add Staff >>>
Id  : S1111
```

```
Name: Alina Amin
Date [dd-mm-yyyy]: 2-6-2020
Time [hour:min] using 24-hour format: 14:00

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 2

<<< Add Product >>>
Type (Promotion/ Regular): Promotion
Name: Casio FX-570MS
Price: RM40

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 2

<<< Add Product >>>
Type (Promotion/ Regular): Regular
Name: Stabilo Swans 12 Colours
Price: RM10.40
Category: Colour Pencil

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 2

<<< Add Product >>>
Type (Promotion/ Regular): Promotion
Name: Artline Whiteboard Marker 500A
Price: RM19.90

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 2

<<< Add Product >>>
Type (Promotion/ Regular): Regular
Name: UHU All Purpose Adhesive Glue 20ml
Price: RM19.90
Category: Glue

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
```

```
[5] Exit
==========================

Select task: 2

<<< Add Product >>>
Type (Promotion/ Regular): Promotion
Name: Smiggle Pencil Case
Price: RM45.90

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 3

<<< Add Sales >>>

Staff list
1) S1111

Select staff: 1

--- Enter Sale Info ---
1) Casio FX-570MS
2) Stabilo Swans 12 Colours (Colour Pencil)
3) Artline Whiteboard Marker 500A
4) UHU All Purpose Adhesive Glue 20ml (Glue)
5) Smiggle Pencil Case

Select product: 2
Quantity: 3

Press 'Y' to continue >> y

--- Enter Sale Info ---
1) Casio FX-570MS
2) Stabilo Swans 12 Colours (Colour Pencil)
3) Artline Whiteboard Marker 500A
4) UHU All Purpose Adhesive Glue 20ml (Glue)
5) Smiggle Pencil Case

Select product: 5
Quantity: 2

Press 'Y' to continue >> n

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 1

<<< Add Staff >>>
Id  : S2233
Name: Alias Zakaria
Date [dd-mm-yyyy]: 3-6-2020
Time [hour:min] using 24-hour format: 08:30

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
```

```
[5] Exit
==========================

Select task: 4

<<< Staff(s) and Sales Info >>>

Number of Staffs: 2

Staff #1
Id   : S1111
Name : Alina Amin
Date : 02-06-2020
Time : 02:00 pm

Number of All Sales    : 2
Number of Promo Sales  : 1
Number of Regular Sales: 1

No  Description                          Unit Price(RM)  Discount(RM)  Quantity  Total(RM)
1.  Stabilo Swans 12 Colours (Colour Pencil)      10.40                     3        31.20
2.  Smiggle Pencil Case                           45.90         6.88        2        78.03

Total sales = RM109.23

Staff #2
Id   : S2233
Name : Alias Zakaria
Date : 03-06-2020
Time : 08:30 am

No sales data!

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 3

<<< Add Sales >>>

Staff list
1) S1111
2) S2233

Select staff: 2

--- Enter Sale Info ---
1) Casio FX-570MS
2) Stabilo Swans 12 Colours (Colour Pencil)
3) Artline Whiteboard Marker 500A
4) UHU All Purpose Adhesive Glue 20ml (Glue)
5) Smiggle Pencil Case

Select product: 3
Quantity: 2

Press 'Y' to continue >> y

--- Enter Sale Info ---
1) Casio FX-570MS
2) Stabilo Swans 12 Colours (Colour Pencil)
3) Artline Whiteboard Marker 500A
4) UHU All Purpose Adhesive Glue 20ml (Glue)
5) Smiggle Pencil Case

Select product: 4
Quantity: 1

Press 'Y' to continue >> n
```

```
========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 3

<<< Add Sales >>>

Staff list
1) S1111
2) S2233

Select staff: 1

--- Enter Sale Info ---
1) Casio FX-570MS
2) Stabilo Swans 12 Colours (Colour Pencil)
3) Artline Whiteboard Marker 500A
4) UHU All Purpose Adhesive Glue 20ml (Glue)
5) Smiggle Pencil Case

Select product: 3
Quantity: 1

Press 'Y' to continue >> y

--- Enter Sale Info ---
1) Casio FX-570MS
2) Stabilo Swans 12 Colours (Colour Pencil)
3) Artline Whiteboard Marker 500A
4) UHU All Purpose Adhesive Glue 20ml (Glue)
5) Smiggle Pencil Case

Select product: 5
Quantity: 3

Press 'Y' to continue >> n

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 4

<<< Staff(s) and Sales Info >>>

Number of Staffs: 2

Staff #1
Id   : S1111
Name : Alina Amin
Date : 02-06-2020
Time : 02:00 am

Number of All Sales    : 4
Number of Promo Sales  : 3
Number of Regular Sales: 1

No  Description                          Unit Price(RM)  Discount(RM)  Quantity  Total(RM)
1.  Stabilo Swans 12 Colours (Colour Pencil)      10.40                     3        31.20
2.  Smiggle Pencil Case                           45.90         6.88        2        78.03
3.  Artline Whiteboard Marker 500A                19.90         2.98        1        16.91
4.  Smiggle Pencil Case                           45.90         6.88        3       117.05

Total sales = RM243.19

Staff #2
```

```
Id   : S2233
Name : Alias Zakaria
Date : 03-06-2020
Time : 08:30 am

Number of All Sales    : 2
Number of Promo Sales  : 1
Number of Regular Sales: 1

No  Description                              Unit Price(RM)  Discount(RM)  Quantity  Total(RM)
1.  Artline Whiteboard Marker 500A                  19.90          2.98         2       33.83
2.  UHU All Purpose Adhesive Glue 20ml (Glue)       19.90                       1       19.90

Total sales = RM53.73

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 5
Thank you! :)
```

(a) Example output of the first run

```
========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 1

<<< Add Staff >>>
Id  : S1111
Name: Alina Hasan
Date [dd-mm-yyyy]: 2-6-2020
Time [hour:min] using 24-hour format: 14:20

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 2

<<< Add Product >>>
Type (Promotion/ Regular): Promotion
Name: Casio AS-120
Price: RM25.60

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 2

<<< Add Product >>>
Type (Promotion/ Regular): Regular
Name: Faber-Castell 0.5mm Blue
```

```
Price: RM27
Category: Ball Pen

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 2

<<< Add Product >>>
Type (Promotion/ Regular): Promotion
Name: Swarovski Ballpoint Pen
Price: RM159

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 3

<<< Add Sales >>>

Staff list
1) S1111

Select staff: 1

--- Enter Sale Info ---
1) Casio AS-120
2) Faber-Castell 0.5mm Blue (Ball Pen)
3) Swarovski Ballpoint Pen

Select product: 1
Quantity: 2

Press 'Y' to continue >> y

--- Enter Sale Info ---
1) Casio AS-120
2) Faber-Castell 0.5mm Blue (Ball Pen)
3) Swarovski Ballpoint Pen

Select product: 2
Quantity: 3

Press 'Y' to continue >> y

--- Enter Sale Info ---
1) Casio AS-120
2) Faber-Castell 0.5mm Blue (Ball Pen)
3) Swarovski Ballpoint Pen

Select product: 3
Quantity: 1

Press 'Y' to continue >> n

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================
```

```
Select task: 4

<<< Staff(s) and Sales Info >>>

Number of Staffs: 1

Staff #1
Id   : S1111
Name : Alina Hasan
Date : 02-06-2020
Time : 02:20 pm

Number of All Sales    : 3
Number of Promo Sales  : 2
Number of Regular Sales: 1

No  Description                          Unit Price(RM)  Discount(RM)  Quantity  Total(RM)
1.  Casio AS-120                               25.60         3.84         2        43.52
2.  Faber-Castell 0.5mm Blue (Ball Pen)        27.00                      3        81.00
3.  Swarovski Ballpoint Pen                    159.00        23.85        1        135.15

Total sales = RM259.67

========== Menu ==========
[1] Add Staff
[2] Add Product
[3] Add Sales
[4] Display Sales
[5] Exit
==========================

Select task: 5
Thank you! :)
```

(b) Example output of the second run

Figure 6: Example runs of program

Question 4

Write a complete C++ program based on the UML class diagram given in **Figure 7**. Your program should be able to produce the output shown in **Figure 8**. Implement all the classes with the member variables (attributes) and member functions (methods) specified in the diagram. The purpose of each function is as the name implies, and some of them are further explained below. Note that the definition for the **`TennisShop`** superclass is fully given (refer to **Figure 9**). Write the program based on the following tasks:

(a) Write a subclass **`Racket`** with the following codes:

(i) Define all the member variables of the class.

(ii) Write the codes for a constructor with no argument that will do nothing.

(iii) Write the codes for a constructor with arguments that will initialize all the member attributes for the class, including the superclass's attribute named **`quantity`**.

(iv) Write the codes for the **`display()`** function that will print the brand, name of racket type, price, and quantity of rackets purchased. **Table 1** shows the prices vary according to the type of racket. The function will also print the total price of rackets purchased by calling

`calcPrice()`. **Note:** You must use proper output formatting to generate the output, as shown in **Figure 8**.

Table 1: Racket type and price

Racket Type	Name of Racket Type	Brand	Price (RM)
1	Junior Tennis Racket	Wilson	569.70
2	Beginners Tennis Racket	Yonex	719.70
3	Advanced Tennis Racket	Yonex	1289.70
4	All Tennis Racket	Dunlop	989.70

(v) Write the codes for **`calcPrice()`** that will calculate and return the total price of rackets purchased.

(b) Write a subclass **`Ball`** with the following codes:

(i) Define all the member variables of the class.

(ii) Write the codes for a constructor with no argument that will do nothing.

(iii) Write the codes for a constructor with arguments that will initialize all the member attributes for the class, including the superclass's attribute named **`quantity`**.

(iv) Write the codes for **`calcDisc()`** function that will calculate and return the new price after the discount deducted, where the discount received is 15 percent of the price of the balls.

(v) Write the codes for the **`display()`** function that will print the brand, description of the ball, number of balls per tube, price, discount received, and quantity of a set of balls purchased. **Table 2** shows the prices vary according to the id of the ball. The function will also print the total price of a set of balls purchased by calling **`calcPrice()`**. **Note:** You must use proper output formatting to generate the output, as shown in **Figure 8**.

Table 2: Ball id and price

Ball Id	Ball Description	Brand	Number of balls (per tube)	Price (RM)
1	Roland Garros All Court	Wilson	4	38.40
2	US Open	Wilson	3	28.80
3	Fort Max TP KNLTB	Dunlop	4	81.00
4	Fort All Court	Dunlop	3	33.00

(vi) Write the codes for **`calcPrice()`** that will call **`calcDisc()`** and return the total price of balls purchased after the discount deducted.

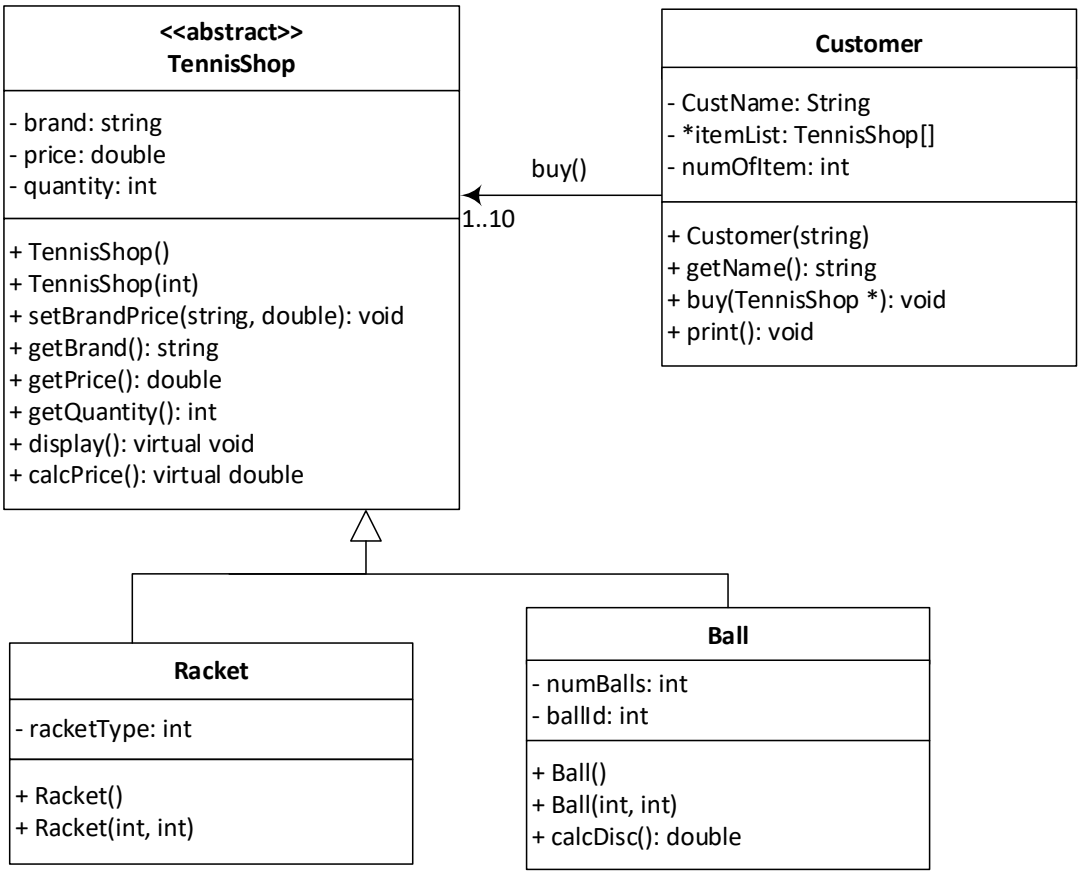

Figure 7: UML class diagram

```
WELCOME ARIF AIMAN

==========[MENU]============
  [1] Buy item
  [2] Print all items
  [3] Exit
============================
Choice: 1

<< Item to buy >>
  [1] Racket
  [2] Ball
Choice: 1

*** Racket ***
  [1] Wilson: Junior Tennis Racket
  [2] Yonex: Beginners Tennis Racket
  [3] Yonex: Advanced Tennis Racket
  [4] Dunlop: All Tennis Racket
Choice: 2
Quantity: 3

==========[MENU]============
  [1] Buy item
  [2] Print all items
  [3] Exit
============================
Choice: 1

<< Item to buy >>
```

```
  [1] Racket
  [2] Ball
Choice: 2

*** Ball ***
  [1] Wilson: Roland Garros All Court
  [2] Wilson: US Open
  [3] Dunlop: Fort Max TP KNLTB
  [4] Dunlop: Fort All Court
Choice: 3
Quantity: 2

==========[MENU]============
  [1] Buy item
  [2] Print all items
  [3] Exit
============================
Choice: 1

<< Item to buy >>
  [1] Racket
  [2] Ball
Choice: 1

*** Racket ***
  [1] Wilson: Junior Tennis Racket
  [2] Yonex: Beginners Tennis Racket
  [3] Yonex: Advanced Tennis Racket
  [4] Dunlop: All Tennis Racket
Choice: 4
Quantity: 1

==========[MENU]============
  [1] Buy item
  [2] Print all items
  [3] Exit
============================
Choice: 2

LIST OF TENNIS ITEMS BOUGHT

----- Item #1 -----
Tennis Racket
Brand   : Yonex
Type    : Beginners Tennis Racket
Price   : RM 719.70
No Discount!
Quantity: 3
Subtotal: RM 2159.10

----- Item #2 -----
Tennis Ball
Brand      : Fort Max TP KNLTB
Description: Dunlop
Price      : RM 81.00 for 4 balls per tube
Discount   : 15.00%
Quantity   : 2
Subtotal   : RM 137.70

----- Item #3 -----
Tennis Racket
Brand   : Dunlop
Type    : All Tennis Racket
Price   : RM 989.70
No Discount!
Quantity: 1
Subtotal: RM 989.70
```

```
TOTAL: RM 3286.50

==========[MENU]============
  [1] Buy item
  [2] Print all items
  [3] Exit
============================
Choice: 4

Invalid input. Try again!

==========[MENU]============
  [1] Buy item
  [2] Print all items
  [3] Exit
============================
Choice: 0

Invalid input. Try again!

==========[MENU]============
  [1] Buy item
  [2] Print all items
  [3] Exit
============================
Choice: 3

Thank you for using this system ;)
```

Figure 8: Example output of the program

(c) Write a class named **`Customer`** with the following codes:

(i) Define all the member variables of the class.

(ii) Write the codes for a constructor with arguments that accepts the customer's name. This value should be assigned to the customer's name member variable. Then, initialize **`numOfItem`** member variable to zero.

(iii) Write the codes for **`getName()`** function to return customer's name.

(iv) Write the codes for **`buy`** function that update the pointers array named **`itemList`** to include the new pointer of **`TennisShop`** in the array (assign an element in the array of **`TennisShop`** pointers with the passed argument). Then, update the number of items inserted into the array.

(v) Write the codes for **`print()`** function that prints the information of the items bought (using **`display()`** function) and update the total payment for all items bought (including discount). **Note:** You must use proper output formatting to generate the output, as shown in **Figure 8**.

(d) Write three functions named **`displayMainMenu()`**, **`displayRacketMenu()`**, and **`displayBallMenu()`** to provide the user a menu-driven interaction. The definition for all the functions are fully given in **Figure 10**.

```
class TennisShop {
 private:
    string brand;
    double price;
    int quantity;

 public:
    TennisShop() {}

    TennisShop(int quantity) {
      this->quantity = quantity;
    }

    void setBrandPrice(string brand, double price) {
      this->price = price;
      this->brand = brand;
    }

    string getBrand() { return brand; }
    double getPrice() { return price; }
    int getQuantity() { return quantity; }

    virtual void display() = 0;
    virtual double calcPrice() = 0;
};
```

Figure 9: `TennisShop` class

```
int displayMainMenu() {
  int choice;
  cout << "==========[MENU]============\n"
       << "  [1] Buy item\n"
       << "  [2] Print all items\n"
       << "  [3] Exit\n"
       << "============================\n"
       << "Choice: ";
  cin >> choice;
  return choice;
}

int displayRacketMenu() {
  int choice;
  cout << "  [1] Wilson: Junior Tennis Racket\n"
       << "  [2] Yonex: Beginners Tennis Racket\n"
       << "  [3] Yonex: Advanced Tennis Racket\n"
       << "  [4] Dunlop: All Tennis Racket\n"
       << "Choice: ";
  cin >> choice;
  return choice;
}

int displayBallMenu() {
  int choice;
  cout << "  [1] Wilson: Roland Garros All Court\n"
       << "  [2] Wilson: US Open\n"
       << "  [3] Dunlop: Fort Max TP KNLTB\n"
       << "  [4] Dunlop: Fort All Court\n"
       << "Choice: ";
  cin >> choice;
  return choice;
}
```

Figure 10: Menu-driven interaction functions

(e) Write **`main()`** function. The definition for **`main()`** function is partially given in **Figure 11**. Complete the function with the following codes:

```
1  int main() {
2    TennisShop *p;
3    Customer c("ARIF AIMAN");
4
5    //Complete your function start from here
6    //Define any suitable variables for the program
7    //Write the codes based on the tasks listed in (e)
8  }
```

Figure 11: Incomplete **`main()`** function

(i) Display welcome message to customer.

(ii) Enter the task chosen.

(iii) In **`case 1`**:

- Enter the item chosen.
- Enter the details of the item chosen.
- Dynamically allocates a new item (racket or ball) object to the **`TennisShop`** pointer.
- Add the item to the **`Customer`** object.

(iv) In **`case 2`**, display the list of the tennis items bought.

(v) In **`case 3`**, display an appropriate message and exit the program.

Figure 8 shows a sample screen output that your program should produce. Note that all the interactions shown in **Figure 8** are continuous in a single run. Note also that the **bold** texts indicate input entered by the user.