

**SCHOOL OF COMPUTING**  
**COMMON INFOCOMM TECHNOLOGY PROGRAMME**  
**DIPLOMA IN APPLIED AI & ANALYTICS**  
**DIPLOMA IN INFOCOMM SECURITY MANAGEMENT**  
**DIPLOMA IN INFORMATION TECHNOLOGY**

**ST0502 Fundamentals of Programming**

2022/2023 SEMESTER 1  
PRACTICAL ASSIGNMENT (CA2)

**Instructions and Guidelines:**

1. CA2 an individual assignment that accounts for 30% of the module. It comprises of 2 parts ie Stage 1 & Stage 2 requirements.
2. You are required to submit softcopy source codes for both Stages 1 & 2 in **BrightSpace** by the deadlines given below.

In your submission, please include a short header with your Name (as in SAS2), Class and Admission number in your program.

**Sample header:**

```
// Name : King Arthur
// Class: DCCC/1A/03
// Adm  : 889977
//
:
:
```

CA2	Submission Deadline in eSP(BrightSpace)	Marks Allocation
Part 1	4 July 2022 (Mon), 8am (Week 12)	20
Part 2	1 Aug 2022 (Mon), 8am (Week 16)	80
	Total	100

3. You must use JavaScript to develop the application in Visual Studio Code.
4. You may consult your Lecturer for overall design and structure of your assignment. However, you are NOT allowed to ask him/her to debug your program. Students who seek debugging help from Lecturer will only be awarded a maximum grade of **C+**. It can be lower depending on the quality of work submitted.
5. You are required to do demo/interview of your programs to your Lecturer individually. You may be asked to explain the program logic during the sessions. Please check with your Lecturer for the detailed schedule.
6. If you are **absent** from the demo/interview **without** a valid Leave of Absence approved by SP, you will be awarded **ZERO mark** even if you have submitted

your assignments. If you are not well on your day of interview and have an approved Leave of Absence (LOA), you have to inform your Lecturer in advance, apply for LOA in Student portal and submit a copy of the approved Leave of Absence by email to your lecturer. Your Lecturer will arrange for a make-up interview session.

7. **50% of marks** to be deducted for submission of assignment within **ONE** calendar day after the deadline. **No marks shall** be awarded for assignments submitted **more than one day** after the deadline.

**For example :** If the submission deadline is 1 Jan, 8am, any submission after that till 2 Jan, 8am will have 50% marks deducted. Any submission from 3 Jan, 8am onwards will NOT be accepted ie given zero.

8. **No marks will be awarded**, if the work is copied, done by someone or you have allowed others to copy your work. This is a **very** serious offence of plagiarism committed by all involved (ie all the givers & recipients). Please refer the clause in **RED** below regarding plagiarism.

Warning: Plagiarism means passing off as one's own the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person. **Plagiarism is a serious offence and disciplinary action** will be taken against you. If you are guilty of plagiarism, you may fail all modules in the semester, or even be liable for expulsion.

## Requirements for Shoppers Membership Application (Pilot)

### 1. Objective

The objective of this assignment is to create a pilot shoppers membership loyalty application.

### 2. Overview

Company XYZ plans to launch a membership loyalty program to boost the sales of its products. The CEO has hired you to create a pilot application with a few features to test out this new idea. These are piloted in two stages i.e. if the first stage is successful, they will proceed to Stage 2. Depending on the outcomes, the CEO will then consider developing the full application.

There are 3 types of memberships: **Diamond, Gold and Ruby**. For each purchase made, loyalty points are awarded based on the amount purchased by the members.

### 3. Stage One Requirements

You are to complete all the Stage 1 requirements below, submit your codes and do a 5~7mins demo to your lecturer.

- 3a. Declare and initialize 5 arrays to represent the 5 different members consisting of the following properties. Subsequently, declare another array – **memberList** to store all the arrays. You may refer to the additional resources provided to perform the aforementioned task.

Name	Leonardo	Catherine	Luther	Bruce	Amy
Membership Type	Gold	Ruby	Gold	Diamond	Ruby
Date Joined	1 Dec 2019	14 Jan 2020	29 Apr 2020	3 Jun 2020	5 Jun 2020
Date of Birth	1 Jan 1980	28 Oct 1985	16 Mar 1992	18 Mar 1994	31 May 2000
Points Earned	1400	250	3350	40200	500

- 3b. Create a menu to display members' details. The output below illustrate the application when executed.

You are to follow the text display as shown in ALL output. You **must** ask for user's name and display it on screen.

The program should allow user to choose the other options without terminating the program. The program should terminate only when **EXIT** is selected.

**Validation of all data input is required.**

Refer to the following screenshots for more details. Text in red boxes denotes user's input. You are **NOT** required to display the **red** box in your output.

**Sample output to prompt for user's name and display the menu options.**

```
Welcome to XYZ Membership Loyalty Programme!
Please enter your name: Alice

Hi Alice, please select your choice:
  1. Display all members' information
  2. Update points earned
  3. Statistics
  4. Exit
>> █
```

**Sample output when an invalid input is entered. Re-prompt user for a valid input.**

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Update points earned
  3. Statistics
  4. Exit
>> 0
Please enter a valid input.

Hi Alice, please select your choice:
  1. Display all members' information
  2. Update points earned
  3. Statistics
  4. Exit
>> █
```

**Sample output when either options (2) or (3) is entered. It displays a message and re-prompt user for input.**

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Update points earned
  3. Statistics
  4. Exit
>> 2
Sorry, work in progress!

Hi Alice, please select your choice:
  1. Display all members' information
  2. Update points earned
  3. Statistics
  4. Exit
>> █
```

**Sample output when option (1) is entered. It displays all members information and re-prompt user for input.**

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Update points earned
  3. Statistics
  4. Exit
```

```
>> 1
```

```
Name: Leonardo
Membership Type: Gold
Date joined: 1 Dec 2019
Date of Birth: 1 Jan 1980
Points Earned: 1400
```

```
Name: Catherine
Membership Type: Ruby
Date joined: 14 Jan 2020
Date of Birth: 28 Oct 1985
Points Earned: 250
```

```
Name: Luther
Membership Type: Gold
Date joined: 29 Apr 2020
Date of Birth: 16 Mar 1992
Points Earned: 3350
```

```
Name: Bruce
Membership Type: Diamond
Date joined: 3 Jun 2020
Date of Birth: 18 Mar 1994
Points Earned: 40200
```

```
Name: Amy
Membership Type: Gold
Date joined: 5 Jun 2020
Date of Birth: 31 May 2000
Points Earned: 500
```

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Update points earned
  3. Statistics
  4. Exit
>> 
```

**Sample output when option (4) is entered, it displays a message and the program is terminated.**

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Update points earned
  3. Statistics
  4. Exit
```

```
>> 4
```

```
Thank you & goodbye!
```

**3c. Assessment Criteria (20marks)**

Marks are allocated based on the following criteria. Students may be asked to explain codes during the demo session.

- Proper declaration of arrays
- Program produces correct output according to requirements
- Data validation is correct and complete
- Program executes properly without crashing/errors produced and terminates properly

## 4. Stage TWO Requirements

After the Company XYZ saw your demonstration, they were very happy with your work but had some suggestions for the application.

First of all, you are to upgrade the current application using the **object oriented concept** learnt in the module. In which, you should not be storing the member's information using arrays but through usage of classes and objects.

Company XYZ was kind enough to suggest that the application should consist of at least 2 of the following classes i.e.

**Member** class to define each member's properties and methods. Properties includes:

- Name
- Membership type (**Diamond, Platinum, Gold or Ruby**)
- Date joined
- Date of birth
- Points earned

**MemberGroup** class to represent an array of all members, other properties and methods.

Feel free to add any properties, other classes or methods you feel are needed in your program.

Hence, the first step was to use objects/classes to create the 5 members consisting their respective properties.

**Validation of all data input is required unless otherwise stated in the requirements.**

- 4a. Next, Company XYZ proposed to increase the choices in the main menu to cater for more options as follows:

Refer to the following screenshots for more details. Text in red boxes denotes user's input. You are **NOT** required to display the **red** box in your output.

**Sample output to prompt for user's name and display the menu options.**

```
Welcome to XYZ Membership Loyalty Programme!  
Please enter your name: Alice  
  
Hi Alice, please select your choice:  
1. Display all members' information  
2. Display member information  
3. Add new member  
4. Update points earned  
5. Statistics  
6. Exit  
>> █
```

- 4b. When the user selects option 2 – **Display member information**, the program will prompt the user to enter the name of the member to display information.

**Sample output when option 2 is entered. Prompt user to enter member's name.**

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
  3. Add new member
  4. Update points earned
  5. Statistics
  6. Exit
>> 2
Please enter member's name: CathERinE

Name: Catherine
Membership Type: Ruby
Date Joined: 14 Jan 2020
Date of Birth: 28 Oct 1985
Points Earned: 250

Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
  3. Add new member
  4. Update points earned
  5. Statistics
  6. Exit
>> █
```

**Sample output when invalid name is entered. Brings user back to main menu.**

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
  3. Add new member
  4. Update points earned
  5. Statistics
  6. Exit
>> 2
Please enter member's name: Charmander
Member does not exist.

Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
  3. Add new member
  4. Update points earned
  5. Statistics
  6. Exit
>> █
```



- 4c. When the user selects option 3 – **Add new member**, the program will prompt user to enter the name of the new member and the date of birth.

**Sample output when option 3 is entered. Prompt user to enter new member's name and date of birth. You may assume the user will enter a valid date of birth ONLY.**

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
  3. Add new member
  4. Update points earned
  5. Statistics
  6. Exit
>> 3
Please enter member's name: Pikachu
Please enter member's date of birth: 1 April 2005

Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
  3. Add new member
  4. Update points earned
  5. Statistics
  6. Exit
>> █
```

**Sample output when name already in database. Re-prompt user to enter a new name.**

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
  3. Add new member
  4. Update points earned
  5. Statistics
  6. Exit
>> 3
Please enter member's name: Amy

Member's name exists in database. Please enter a new name.
Please enter member's name: bRUce

Member's name exists in database. Please enter a new name.
Please enter member's name: █
```

- 4d. When the user selects option 4 – **Update points earned**, the program will prompt user to enter the name of the member and the amount spent. The table below shows the points earned by the member based on the amount spent.

Amount Spent	Points Earned
Less than \$50	10
\$51 to \$100	50
\$101 to \$200	100
\$201 to \$500	200
\$501 to \$1000	500
\$1001 to \$2500	1000
\$2501 and above	2000

**Sample output when option 4 is entered. Prompt user to enter member's name and amount spent.**

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
  3. Add new member
  4. Update points earned
  5. Statistics
  6. Exit
>> 4
Please enter member's name: AMY
Please enter amount spent: 1000

Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
  3. Add new member
  4. Update points earned
  5. Statistics
  6. Exit
>> █
```

**Sample output when name does not exist in database. Prompt user back to the main menu.**

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
  3. Add new member
  4. Update points earned
  5. Statistics
  6. Exit
>> 4
Please enter member's name: K3nn3th
Member does not exist.

Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
```

**NOTE:** The default membership type for all newly joined members is ‘Ruby’. If the member manages to accumulate above certain amount of points, his/her membership type will be upgraded to the next tier. Your program should automatically upgrade the membership type upon reaching the respective points.

Membership Type	Minimum Points to achieve
Diamond	20000
Platinum	5000
Gold	500
Ruby (Default)	0

- 4e. When the user selects option 5 – **Statistics**, the program will display a sub menu shown below.

**Sample output when option 5 is entered. A sub menu is displayed and prompts user for an option.**

```
Hi Alice, please select your choice:
  1. Display all members' information
  2. Display member information
  3. Add new member
  4. Update points earned
  5. Statistics
  6. Exit
>> 5
Please select an option from the sub-menu:
  1. Display names of (all) a certain type of members only.
  2. Display the name of the youngest and oldest member in the system.
  3. Display the name of members with the highest and lowest points earned.
  4. Display total number of members in each membership type.
  5. Display the total points in each membership type.
  6. Return to main-menu
>> █
```

**Sample output when option 1 of the sub menu is entered. Prompt user for Membership Type until a valid Membership Type is entered.**

```
Please select an option from the sub-menu:
  1. Display names of (all) a certain type of members only.
  2. Display the name of the youngest and oldest member in the system.
  3. Display the name of members with the highest and lowest points earned.
  4. Display total number of members in each membership type.
  5. Display the total points in each membership type.
  6. Return to main-menu
>> 1
Enter Membership Type: Sapphire
Please enter a valid membership type.

Enter Membership Type: Diamond
Member(s) of membership type diamond: Bruce.

Please select an option from the sub-menu:
```

**Sample output when option 2 of the sub menu is entered.**

```
Please select an option from the sub-menu:
1. Display names of (all) a certain type of members only.
2. Display the name of the youngest and oldest member in the system.
3. Display the name of members with the highest and lowest points earned.
4. Display total number of members in each membership type.
5. Display the total points in each membership type.
6. Return to main-menu
>> 2
Youngest member: Amy
Oldest member : Leonardo

Please select an option from the sub-menu:
1. Display names of (all) a certain type of members only.
2. Display the name of the youngest and oldest member in the system.
```

**Sample output when option 3 of the sub menu is entered.**

```
Please select an option from the sub-menu:
1. Display names of (all) a certain type of members only.
2. Display the name of the youngest and oldest member in the system.
3. Display the name of members with the highest and lowest points earned.
4. Display total number of members in each membership type.
5. Display the total points in each membership type.
6. Return to main-menu
>> 3
Highest member: Bruce
Lowest member : Catherine

Please select an option from the sub-menu:
1. Display names of (all) a certain type of members only.
2. Display the name of the youngest and oldest member in the system.
```

**Sample output when option 4 of the sub menu is entered. Do note that your result may differ from the sample output due to different test cases involved.**

```
Please select an option from the sub-menu:
1. Display names of (all) a certain type of members only.
2. Display the name of the youngest and oldest member in the system.
3. Display the name of members with the highest and lowest points earned.
4. Display total number of members in each membership type.
5. Display the total points in each membership type.
6. Return to main-menu
>> 4
ruby: 1
gold: 3
platinum: 0
diamond: 1

Please select an option from the sub-menu:
1. Display names of (all) a certain type of members only.
2. Display the name of the youngest and oldest member in the system.
```

**Sample output when option 5 of the sub menu is entered. Do note that your result may differ from the sample output due to different test cases involved.**

```
Please select an option from the sub-menu:
1. Display names of (all) a certain type of members only.
2. Display the name of the youngest and oldest member in the system.
3. Display the name of members with the highest and lowest points earned.
4. Display total number of members in each membership type.
5. Display the total points in each membership type.
6. Return to main-menu
>> 5
ruby: 250
gold: 6250
platinum: 0
diamond: 40200
```

```
Please select an option from the sub-menu:
1. Display names of (all) a certain type of members only.
2. Display the name of the youngest and oldest member in the system.
```

**Sample output when option 6 of the sub menu is entered.**

```
Please select an option from the sub-menu:
1. Display names of (all) a certain type of members only.
2. Display the name of the youngest and oldest member in the system.
3. Display the name of members with the highest and lowest points earned.
4. Display total number of members in each membership type.
5. Display the total points in each membership type.
6. Return to main-menu
>> 6

Hi Alice, please select your choice:
1. Display all members' information
2. Display member information
3. Add new member
4. Update points earned
5. Statistics
6. Exit
>> █
```

#### 4. Assessment Guidelines

Marking criteria will be based on student's ability to demonstrate program requirements and explanation demonstration and Q&A interview in the following areas:

Assessment Criteria	Maximum marks allocated
Program functionalities (Part 1) <ul style="list-style-type: none"> <li>- Execute basic requirements mentioned above correctly</li> <li>- Include appropriate validations</li> </ul>	20
Program functionalities (Part 2) <ul style="list-style-type: none"> <li>- Execute basic requirements mentioned above correctly</li> <li>- Include appropriate validations</li> </ul>	45
Program design: <ul style="list-style-type: none"> <li>- Correct and efficient usage of classes and programming constructs</li> <li>- Appropriate method decomposition</li> <li>- Proper use of Arrays, Functions, Loops, if-else etc</li> <li>- Code efficiency</li> </ul> Program readability: <ul style="list-style-type: none"> <li>- Meaningful identifiers</li> <li>- Meaningful comments and indentation in source documentation</li> </ul>	20
Innovation and creativity or any advanced features	15
Total	100

Validate dates

-- End --