SINGAPORE UNIVERSITY OF SOCIAL SCIENCES

# ICT 133
# Structured Programming

---

## Tutor-Marked Assignment

## July 2023 Presentation

---

## *TUTOR-MARKED ASSIGNMENT (TMA)*

This assignment is worth 24 % of the final mark for ICT133, Structured Programming.

The cut-off date for this assignment is **Sunday, 15 October 2023, 2355 hours**.

Assignment Requirements:

- Do NOT define classes for this TMA.
- **Unless specified in the question**, **you CANNOT use packages not covered in this module, e.g., re, collections, numpy, pandas etc.**
- All functions must be documented. Provide sufficient comments to your code and ensure that your program adheres to good programming practices such as not using global variables.
- **Do not use the exit() function.**
- Failing to do so can incur a penalty of as much as 50% of the mark allotted.

Submission Details:

- Use the template word document provided - **SUSS_PI_No-FullName_TMA.docx.** Rename the file with your SUSS PI and full name join with "_TMA" e.g., "**SUSS_PI-TomTanKinMeng_TMA.docx**" (without the quotes).
- Include the following particulars on the first page of the word document, on separate lines: **Course Code, SUSS PI No., Your Name, Tutorial Group and Submission Date**.
- Copy and paste the source code of each program you write in **text** format. Submit screenshots for **only** output of your program, where applicable.
- If you submit the source code as a screenshot, your code will not be marked. That is, **screenshot code will be awarded zero mark.**
- Submit your solution in the form of a **single MS Word document. Do NOT submit as a pdf document**. You will be penalised if you fail to submit a word document.
- The word document must be submitted to your respective T group. Besides this, you are required to upload your source code to Vocareum.

*Answer all questions. (Total 100 marks)*

**Question 1 (30 marks)**

This question covers materials in Seminar 1 and 2. Use and express selection structure for this question. Functions, repetition, or collections are not necessary for this question.

Company J-Hoop uses the following tables to help online customers determine their basketball shoe size.

**Men's shoe sizing guide**

| **US** | **6** | **6.5** | **7** | **7.5** | **8** | **8.5** | **9** | **9.5** | **10** | **10.5** | **11** | **11.5** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Foot Length** | 24.0 | 24.5 | 25.0 | 25.5 | 26.0 | 26.5 | 27.0 | 27.5 | 28.0 | 28.5 | 29.0 | 29.5 |

All measurements are in centimetres

**Ladies' shoe sizing guide**

| **US** | **4.5** | **5** | **5.5** | **6** | **6.5** | **7** | **7.5** | **8** | **8.5** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Foot Length** | 21.5 | 22.0 | 22.5 | 23.0 | 23.5 | 24.0 | 24.5 | 25.0 | 25.5 | 26.0 |

**Table 1-1**

(a) Write a program that helps online customers find their basketball shoe size.

Your program will read in the customer's gender and foot length measurement in centimetres. Based on Table 1-1, determine the appropriate shoe size. However, there are exceptions:

- For foot length beyond 29.5cm or smaller than 21.1cm, J-Hoop will need to custom-made a pair for the customer. Display appropriate message in this case.
- For men with foot length smaller than 23.6cm, look into the Ladies' shoe sizing guide and recommend accordingly.
- For ladies with foot length larger than 26.0cm, look into the Men's shoe sizing guide and recommend accordingly.

Assume user will enter valid input. Study the sample executions below carefully:

| Sample run 1: | Comments |
|---|---|
| Enter gender (M/F): **m** | |
| Enter foot length (cm): **27.2** | Men's US 9.5 can accommodate up to 27.5cm |
| Your shoe size is Men's US 9.5 | |
| Sample run 2: | |
| Enter gender (M/F): **F** | |
| Enter foot length (cm): **24.0** | Ladies US 7 can accommodate up to 24.0cm |
| Your shoe size is Ladies US 7 | |
| Sample run 3: | |
| Enter gender (M/F): **M** | |
| Enter foot length (cm): **29.8** | Recommend custom-made as no shoe on the shelves can fit the customer |
| Call Customer Service for custom-made shoe | |

| | |
|---|---|
| Sample run 4:<br>Enter gender (M/F): **f**<br>Enter foot length (cm): **20.2**<br>Call Customer Service for custom-made shoe | Recommend custom-made as no shoe on the shelves can fit the customer |
| Sample run 5:<br>Enter gender (M/F): **M**<br>Enter foot length (cm): **23.1**<br>You can try Ladies' US 6.5 | The foot length is less than 23.6cm. Hence, look into Ladies shoe sizing and US 6.5 can accommodate up to 23.5cm |
| Sample run 6:<br>Enter gender (M/F): **F**<br>Enter foot length (cm): **26.8**<br>You can try Men's US 9 | The foot length is more than 26.0cm. Hence, look into Men's shoe sizing and US 9 can accommodate up to 27.0cm |

(20 marks)

(b) As online ordering has increased in folds since the pandemic, the number of returns and exchanges also increased, due to wrong fittings. In order to provide more accurate/better fittings, J-Hoop has requested to include width measurement to help determine if customer foot is narrow or wide.

| Foot | Wide | Standard | Narrow |
|---|---|---|---|
| Length / Width | < 2.35 | 2.35 – 2.65 | > 2.65 |
| Recommendation | next larger size | - | next smaller size |

**Table 1-2**

Using Table 1-2, enhance your program in Q1(a) to recommend the shoe size, based on additional input of customers' foot width measurement.

Assume user will enter valid input. Study the sample executions below carefully:

| Sample run 1:<br>Enter gender (M/F): **m**<br>Enter foot length (cm): **27.2**<br>Enter foot width (cm): **10.5**<br>Your shoe size is Men's US 9.5 | Comments<br>Foot is standard width, and men's US 9.5 can accommodate up to 27.5cm |
|---|---|
| Sample run 2:<br>Enter gender (M/F): **F**<br>Enter foot length (cm): **23.8**<br>Enter foot width (cm): **10**<br>Your shoe size is Ladies US 7 | Foot is standard width, and ladies US 7 can accommodate up to 24.0cm |
| Sample run 3:<br>Enter gender (M/F): **m**<br>Enter foot length (cm): **27.2**<br>Enter foot width (cm): **11.8**<br>Recommended shoe size is Men's US 10 | Men's US 9.5 can accommodate up to 27.5cm, but foot is wide. Hence, recommends next larger size |
| Sample run 4: | |

| | |
|---|---|
| Enter gender (M/F): **F**<br>Enter foot length (cm): **23.8**<br>Enter foot width (cm): **10.3**<br>Recommended shoe size is Ladies US 7.5 | Ladies US 7 can accommodate up to 24.0cm, but foot is wide. Hence, recommends next larger size |
| Sample run 5:<br>Enter gender (M/F): **m**<br>Enter foot length (cm): **27.2**<br>Enter foot width (cm): **10**<br>Recommended shoe size is Men's US 9 | Men's US 9.5 can accommodate up to 27.5cm, but foot is narrow. Hence, recommends next smaller size |
| Sample run 6:<br>Enter gender (M/F): **F**<br>Enter foot length (cm): **23.8**<br>Enter foot width (cm): **8.9**<br>Recommended shoe size is Ladies US 6.5 | Ladies US 7 can accommodate up to 24.0cm, but foot is narrow. Hence, recommends next smaller size |
| Sample run 7:<br>Enter gender (M/F): **f**<br>Enter foot length (cm): **21.1**<br>Enter foot width (cm): **7.7**<br>Call Customer Service for custom-made shoe | Ladies US 4.5 can accommodate up to 21.5cm, but foot is narrow. Hence, recommends next smaller size, which is not available. Hence custom-made |

(10 marks)

For each part, submit separate python codes (otherwise, you will not get full credits) and paste screenshots of at least three program executions, with different input.

**Question 2 (30 marks)**

This question covers materials up to Seminar 3. Make use of functions, selection, and repetition structures. **NO data structures like sets, lists or dictionary should be used for this question.** Keep the program modular by defining other functions if necessary.

**This question is divided into 3 parts. Submit separate python codes for each part. Paste screenshot of a program executions that covers all scenarios.**

(a) Write a Python program that simulate the self-ordering & checkout terminal for a Café.

The Café has the following food item in its menu:

| Code | Item | Price |
|---|---|---|
| A | Soup of the day | $4.50 |
| B | Garden Salad | $5.50 |
| C | BLT Sandwich | $6.50 |

Your program should present the above menu before allowing customer to enter item codes to purchase, until the customer enters "X" or "x" to stop.

The total price will then be computed and displayed.

During the program execution, remember to validate the customer entry of item codes, sentinel values and etc.

A sample program execution is as follows:

```
<< Café Menu >>
A.   Soup of the day
B.   Garden Salad
C.   BLT Sandwich
X.   Exit
Enter your order: A
Enter your order: C
Enter your order: B
Enter your order: A
Enter your order: X
Thank you, please pay $21.00
```

(10 marks)

(b) Enhance Q2(a) to offer 10% discount to Café members if the total price is more than $20. Your program should now:

- first, display the total price
- then only ask for membership if the total price is more than $20
- if the customer is a member, apply the 10% discount

| Sample run 1: | Sample run 2: |
|---|---|
| << Café Menu >><br>A. Soup of the day<br>B. Garden Salad<br>C. BLT Sandwich<br>X. Exit<br>Enter your order: **A**<br>Enter your order: **C**<br>Enter your order: **B**<br>Enter your order: **A**<br>Enter your order: **B**<br>Enter your order: **X**<br>Total: $26.50<br>Are you a member? (Y/N): **y**<br>Thank you, please pay $23.85 | << Café Menu >><br>A. Soup of the day<br>B. Garden Salad<br>C. BLT Sandwich<br>X. Exit<br>Enter your order: **A**<br>Enter your order: **C**<br>Enter your order: **b**<br>Enter your order: **a**<br>Enter your order: **B**<br>Enter your order: **X**<br>Total: $26.50<br>Are you a member? (Y/N): **N**<br>Thank you, please pay $26.50 |

(10 marks)

(c) Enhance Q2(b) to cater to the following combo discounts given to all customers:

- for every A and B ordered, this combo pricing is $9.50
- for every A, B and C ordered, this combo pricing is $15.00

After applying the above combo discounts, your program will continue to offer 10% discount to Café members if the total price is more than $20. In other words, apply part (b) logic.

| Sample run 1: | Sample run 2: |
|---|---|
| *<< Café Menu >>*<br>*A. Soup of the day*<br>*B. Garden Salad*<br>*C. BLT Sandwich*<br>*X. Exit*<br>*Enter your order:* ***A***<br>*Enter your order:* ***C***<br>*Enter your order:* ***B***<br>*Enter your order:* ***A***<br>*Enter your order:* ***B***<br>*Enter your order:* ***X***<br>*Total: $24.50*<br>*Are you a member? (Y/N):* ***Y***<br>*Thank you, please pay $22.05* | *<< Café Menu >>*<br>*A. Soup of the day*<br>*B. Garden Salad*<br>*C. BLT Sandwich*<br>*X. Exit*<br>*Enter your order:* ***A***<br>*Enter your order:* ***C***<br>*Enter your order:* ***B***<br>*Enter your order:* ***A***<br>*Enter your order:* ***B***<br>*Enter your order:* ***X***<br>*Total: $24.50*<br>*Are you a member? (Y/N):* ***n***<br>*Thank you, please pay $24.50* |

(10 marks)

**Question 3 (40 marks)**

This question covers materials up to seminar 4. The data structure to use is List. You can use more than ONE list or nested list. No dictionary collection is required for this question. Keep the program modular by defining necessary functions.

(a) Employ structured programming to develop a simple arithmetic scrabble game, played by minimum of 2 players.

Each players draws a hand (number) of digits and take turns to create correct expressions that match a random value for that round.

- An expression consists of 3 letters 9?9, where the first and third letters are digits, and the second letter ? can only be + or - .
- Digit 0 is a joker and when used in an expression will always yield the correct result.

The first player to clear all the digits wins the game.

Illustration of the game play on left column, with rules of the game explained on the right column:

| | |
|---|---|
| Enter player: **Alan** | |
| Enter player: **<Enter>** | |
| Minimum 2 players | Min 2 players |
| Enter player: **alan** | |
| No duplicate name… re-enter | No duplicate names |
| Enter player: **Betty** | |
| Enter player: **<Enter>** | Computer randomized the play sequence and will stay the same for every game/round |
| Starting hand (number) of digits: **3** | |
| This number must be even and > 3 … | Starting hand must be even and more than 3 |
| Starting hand (number) of digits: **4** | |

| | |
|---|---|
| Let’s play… | Each player is assigned 4 random digits (0-9) before a round begins. In this case, Alan is given [5,2,1,1] and Betty has [2,0,9,5] |
| Round 1: Result 6 | At start of round, computer will generate a random number (1-9) as the “result” |
| Alan’s hand: [5, 2, 1, 1]<br>Enter expression: **5+1**<br>Correct!<br>Alan’s hand: [2, 1] | Alan can use 5, 1 from his hand to form “5+1” |
| Betty’s hand: [2, 0, 9, 5]<br>Enter expression: **9+0**<br>Correct!<br>Betty’s hand: [2, 5] | Any expression with 0 and proper format is correct. |
| Round 2: Result 5 | |
| Alan’s hand: [2, 1]<br>Enter expression: **<Enter>**<br>Skipped<br>Alan’s hand: [2, 1] | Hit <Enter> key to skip if cannot form an expression. |
| Betty’s hand: [2, 5]<br>Enter expression: **5-**<br>Incorrect!!<br>Betty’s hand: [2, 5] | Expression is not in proper format (3 letters of which the first and third must be digits, and the operator in between must be + or -) |
| Round 3: Result 3 | |
| Alan’s hand: [2, 1]<br>Enter expression: **2+1**<br>Correct!<br>Alan wins this game in 3 rounds!! | No more digits in hand. Alan wins this game! |

(25 marks)

(b) Modify the program in Q3(a), with the following new rules:

- Players will need to win 3 games to become overall winner.
- Implement penalty – for player who skipped, entered incorrect expression, or entered expression that evaluated to wrong result, will draw 2 random digits into their hand.
- Player who loses the game will be awarded a “Skip” card in the next game. With the “Skip” card, player can hit <Enter> to skip, without penalty.
- If not utilised, the “Skip” cards CANNOT be carried forward to next game.

Illustration of the "new" game on left column, with rules of the game explained on the right column:

| | |
|---|---|
| Enter player: **Alan**<br>Enter player: **Betty**<br>Enter player: **<Enter>**<br><br>Starting hand (number) of digits: **4** | |
| Let's play…Game 1 | Computer randomized the play sequence and will stay the same for every game/round |
| Round 1: Result 6<br><br>Betty's hand: [5, 2, 1, 1]<br>Enter expression: **5+1**<br>Correct!<br>Betty's hand: [2, 1]<br>…<br>…<br>… | |
| Alan wins this game in 4 rounds!!<br><br>Overall game score: Alan 1, Betty 0<br><br>Let's play…Game 2<br><br>Round 1: Result 5 | End of game, if no overall winner, continue to play another game. Loser will be awarded a "Skip" card in next game. |
| Betty's hand: [3, 6, 5, 7, 'S']<br>Enter expression: **<Enter>** | 'S' represents the "Skip" card awarded to Betty |
| Skipped with no penalty!!<br>Betty's hand: [3, 6, 5, 7]<br><br>Alan's hand: [2, 1, 2, 1] | No penalty for Betty as she has a "Skip" card |
| Enter expression: **<Enter>**<br>Skipped and 2 digits added to hand<br>Alan's hand: [2, 1, 2, 1, 4, 7]<br>…<br>…<br>… | Hit <Enter> key to skip if cannot form an expression. In this case, player who skipped will draw 2 random digits into their hand |
| Betty wins this game in 5 rounds!!<br><br>Overall game score: Alan 1, Betty 1<br><br>Let's play…Game 3<br><br>Round 1: Result 6<br><br>Betty's hand: [1, 2, 8, 7]<br>Enter expression: **8-2**<br>Correct!<br>Betty's hand: [1, 7] | End of game, if no overall winner, continue to play another game. Loser will be awarded a "Skip" card in next game. |

| | |
|---|---|
| Alan’s hand: [3, 7, 4, 1, ‘S’] | ‘S’ represents the “Skip” card awarded to Alan |
| Enter expression: **4+1** | |
| Incorrect and 2 digits added to hand<br>Alan’s hand: [3, 7, 4, 1, ‘S’, 1, 2] | PENALTY – draw 2 digits if expression cannot evaluate to match result |
| …<br>…<br>… | |
| Betty wins this game in 2 rounds!!<br>Overall game score: Alan 1, Betty 2<br>Let’s play…Game 4 | End of game, if no overall winner, continue to play another game. Loser will be awarded a “Skip” card in next game. |
| Round 1: Result 5 | |
| Betty’s hand: [3, 9, 5, 8]<br>Enter expression: **8-3**<br>Correct!<br>Betty’s hand: [9, 5] | |
| Alan’s hand: [6, 2, 5, 5, ‘S’]<br>Enter expression: **5+**<br>Incorrect and 2 digits added to hand<br>Alan’s hand: [6, 2, 5, 5, ‘S’, 0, 0] | PENALTY – draw 2 digits if expression is not in proper format (3 letters of which the first and third must be digits, and the operator in between must be + or -) |
| Round 2: Result 4 | |
| Betty’s hand: [9, 5]<br>Enter expression: **9-5**<br>Correct!<br>Betty’s hand: [9, 5] | |
| Betty wins this game in 2 rounds!! | |
| Overall game score: Alan 1, Betty 3<br>Betty is the overall winner!! | Game stop when any player reaches 3 |

(15 marks)

For each part, submit separate python codes (otherwise, you will not get full credits) and paste screenshots of at least three program executions, with different outcomes.

**---- END OF ASSIGNMENT ----**