UNIT SAINS KOMPUTER

KOLEJ MATRIKULASI LABUAN

CP115
ASSIGNMENT

Problem Statement

You are tasked with creating a simple Python program to determine the eligibility of individuals for discounts at a movie theater based on their age. The theater offers the following discount categories:

- **Children** (age 12 and below): 50% discount
- **Teenagers** (age 13 to 17): 25% discount
- **Adults** (age 18 and above): No discount

Write a Python program that prompts the user to enter their age and the price of the movie ticket. Based on the provided age and ticket price, the program should output the appropriate discount category for the user and calculate the discounted price for the movie ticket.

Requirements

1. Your program should prompt the user to enter their age as an integer.
2. Prompt the user to enter the price of the movie ticket as a floating-point number.
3. Output the discount category as a string (e.g., "Children", "Teenagers", "Adults") along with any applicable discount percentage.
4. Calculate the discounted price for the movie ticket based on the user's age, the corresponding discount category, and the entered ticket price.
5. Handle invalid inputs gracefully. If the user enters a non-numeric value, a negative age, or a negative ticket price, display an error message and terminate the program.

Example Input and Output

```
None
Enter your age: 15

Enter the price of the movie ticket: 10.00

You are eligible for the Teenagers discount (25% off).

Discounted ticket price: $7.50
```

Submission Guidelines

The assignment will have to be submitted on Week 15 on a Google Forms that will be released later.

Required Submission

1. Complete Documentation (PDF Format)

- Submit a single PDF document named assignment_CP115.pdf containing:
    - **Cover Page**: Include your name, matric number and assignment title
    - **IPO Table**
    - Detailed **pseudocode** for the solution
    - Complete **flowchart** diagram showing program logic
    - Well-commented **Python code** with proper formatting
    - Screenshots and analysis of program execution with at least 10 different test cases, including:
        - Valid inputs for each age category (children, teenagers, adults)

        - Invalid inputs (negative values, non-numeric entries)
        - Edge cases (boundary ages: 12, 13, 17, 18)
    - **Reflection**: Brief summary of challenges faced and lessons learned

2. Python Source File(.py file)

- Submit your working Python program as discount_calculator.py
- Code must be properly commented and follow Python naming conventions
- Include appropriate error handling as specified in the requirements.

Rubric

Criteria	Missing Item	Low	Average	Excellent	Weight	Mark		Total
	0	1	2	3		Examiner	Moderator	
Problem Analysis ● Input ● Process ● Output	Unable to organize and analyze gathered data or information and fails to define the root of the problem.	Finds difficulty in organizing and analyzing gathered data or information and finds difficulty in describing the root of the problem.	Able to organize and analyze gathered data or information but does not clearly describe the root of the problem.	Excellent organizing and analyzing of gathered data or information and clearly describe the root of the problem.	1			
	All Input, Process, Output are incorrect.	At least one correct.	At least 2 correct.	All Input, Process, Output are correct.				
Design a solution. ● Pseudocode ● Flowchart	Unable to provide algorithm to represent solution of given problem.	Able to provide algorithm partially and does not fully represent the solution of given problem.	Able to provide algorithm but does not fully represent the solution of given problem.	Able to provide clear and accurate algorithm to logically represent the solution of given problem.	2			
	No algorithm is provided.	● Provide both algorithms (pseudocode and flowchart). ● Both algorithms contain error(s).	● Provide both algorithms (pseudocode and flowchart). ● One of the algorithms contain error(s).	Both algorithms are correct.				

Criteria	Missing Item	Low	Average	Excellent	Weight	Mark		Total
	0	1	2	3		Examiner	Moderator	
Implementation ● Python Program	No program is provided.	The program is developed but with syntax error.	The program is developed and runs successfully but does not fully solve the given problem.	The program is successfully developed based on presented algorithm and run successfully.	2			
	No program.	Contain syntax error.	Contain runtime/ logic error.	Program runs successfully and solve the given problem.				
Testing and Verification	Did not provide any sample output.	Provide only one sample output.	Provide at least two sample output.	Provide all possibilities of test data with complete sample output.	1			
	No sample of output.	The sample covers one possibility only.	The sample covers two possibilities only.	Covers all possibilities.				
Originality	There is evidence that shows the task is totally copy of other people's work	Only a few parts of the task show the original work.	Most of the task shows original work.	The task shows substantial originality.	1			

Criteria	Missing Item	Low	Average	Excellent	Weight	Mark		Total
	0	1	2	3		Examiner	Moderator	
	(from any source).							
	Totally copy of other people's work.	Perform a task with limited trust, honesty, sincerity, and transparency.	Most of the task shows original work.	Always perform a task with trust, honesty, sincerity and transparent in any situation.				
Documentation Comments and Descriptions	No comments or description provided.	Only a part of comments/descripti on provided.	Comments/ descriptions completely provided but unclear.	The comments/ descriptions are complete, with clear explanation reflecting well understanding of task.	1			
	No comments/ descriptions are provided at all.	Very minimum comments/ descriptions provided.	Comments/ descriptions provided but not sufficient to describe the program.	Comments/ descriptions are sufficiently describing the program.	**Total**			