Royal University of Bhutan

Continuous Practical Assignment I

Programming Methodology (CSF101)

Bachelors of Engineering in Electrical Engineering (EE), Mechanical Engineering (ME), Instrumentation and Control Engineering (ICE) & Electronics and Communication Engineering (ECE)

Objective:

1. **Programming Skills:** To demonstrate your proficiency in Python programming, including knowledge of data types, control structures, functions, and libraries/modules relevant to automation tasks.
2. **Problem-Solving:** To solve a real-world problem or automate a specific task using Python. This encourages students to apply their programming skills to practical scenarios.
3. **Creativity and Innovation:** Encouragement to think creatively and innovatively to come up with unique solutions by using the concepts learnt in class.

Problem Statement:

In a rock-paper-scissors game:

- Rock defeats Scissors
- Scissors defeats Paper
- Paper defeats Rock.
- Same shape = draw.

Score system:
The score for the shape you selected (1 for Rock, 2 for Paper, and 3 for Scissors) + the score for the outcome of the round (0 if you lost, 3 if the round was a draw, and 6 if you won).
Your total score is the sum of your scores for each round.

Example of the scoring system:
Round 1: You choose rock (1), and your opponent chooses scissors; You win(6); the score is 1 + 6 = 7
Round 2: You choose paper (2), and your opponent chooses scissors; You lose(0); the score is 2 + 0 = 2
Round 3: You choose paper (2), your opponent chooses paper; You draw(3); the score is 2 + 3 = 5

Total score: 7 + 2 + 5 = 14

Your Example Input:

```
A Y
B X
C Z
```

The first column is what your opponent is going to play: A for Rock, B for Paper, and C for Scissors. Each line is a round.
The second column says how the round needs to end: X means you need to lose, Y means you need to end the round in a draw, and Z means you need to win. *Calculate what you need to draw to satisfy the round.*

Based on the example input:

1. In the first round, your opponent will choose Rock (A), and you need the round to end in a draw (Y), so you also choose Rock. This gives you a score of 1 + 3 = 4.
2. In the second round, your opponent will choose Paper (B), and you choose Rock so you lose (X) with a score of 1 + 0 = 1.
3. In the third round, you will defeat your opponent's Scissors with Rock for a score of 1 + 6 = 7.

Your total score is 4 + 1 + 7 = 12

Instructions:

1. **Take your input file:**
    - The input file is the last number of your student number. If your student number ends with 9, then your input file is "input_9_cap1.txt"
    - You may find your input file from the link here.
2. **Read the input file using python:**
    - Write a python code to read all lines from your input file.

Refer here to learn how to read the file .

3. **Calculate the total score:**
    - Based on your input file, write a python program to calculate the total score by summing all of the scores of the rounds.

Note:

- ***PLAGIARISM WILL NOT BE TOLERATED*** and will result in 0 mark. [WAL Section D4: Academic Dishonesty].
- Any internet articles you refer while solving the Assignment must be put at the top of the code in comments.
- Follow the submission guidelines. Failure to follow the instructions will result in deduction of marks or *0 if none of the instructions are followed.*
- Write your solution python code in **one file.** The structure of the solution file is detailed below.
- Your code should be explained in a detailed manner using comments wherever necessary.

Submission:

1. Create a GitHub repository named "CSF101 CAP" on your profile and keep it "public"
2. Upload your CAP1 solution to the aforementioned github repository.
3. Submit your solution file link in **VLE**.
4. Your solution program/code file should be named as: CAP1_ID.PY

    Example:
    CAP1_02233456.py

5. Submit before 7th April, 2024. *The VLE submission will close on midnight 7th April.*

Solution File Structure:

CAP1_02123456.py

```
################################
# Your Name
# Your Section
# Your Student ID Number
################################
# REFERENCES
# Links that you referred while solving
# the problem
# http://link.to.an.article/video.com
################################
# SOLUTION
# Your Solution Score:
# Put your number here
################################

# Read the input.txt file
def read_input():
   # code here to read your input file

# solution
def calculate_score():
   # implement your solution here.
   # print your solution to output as: "the total score is: <score>"

# Other parts of code here to run your functions and printing of the input.

# Note: You may add parameters/arguments, return values to the functions above.
```

Marking Rubrics

Rubric for RA Practical Assignment I

Criteria	Description	Weightage	Accomplished (3)	Good (2)	Satisfactory (1)	Needs Improvement (0)
Correctness	Is the solution score provided correct	10%	Solution score is completely accurate	-	-	Solution score is completely wrong.
Code Quality & Documentation	Comments Proper Variable Names Solution file is not cryptic.	10%	Code is well organized, documented and uses good naming conventions.	Reasonably good coding practices followed, some room for improvement	Code is disorganized or lacks documentation and structure.	No variable naming convention, no documentation, Unable to understand code.
Reading Input.txt	Implementation of read_input() function.	30%	Reads input.txt file properly, Parses the string, Proper string cleaning.	Reads input.txt file, String parsing and cleaning. Some room for improvement	Reads input.txt file, No string parsing or cleaning	Fails to read input.txt file,
Score Calculation Implementation	Implementation of calculate_score()	50%	Solution Algorithm is correct and generates solution as per instruction	Algorithm is correct, doesn't output the solution as instructed	Algorithm incorrect but explains logic and thinking process well in comments and code.	No solution is generated and no algorithm implemented in code/comments

Note: 100 marks converted to 15 mark