University of South Australia

UniSA STEM

COMP 1039 Problem Solving and Programming

Practical 6

Create a separate file in order to complete each of the following exercises (refer to practical 1 or ask your supervisor if you are having problems doing so). Your files should include appropriate comments consisting of the following – file name, your details, version number and date, brief program description and the University's academic misconduct statement.

Question 1 - A game of rock-paper-scissors (part a):

The following exercise provides you with the opportunity to implement your solution to an earlier problem solving workout exercise (if you haven't already done so).

1. Write a program that plays a single game of rock-paper-scissors against the computer. The program should:
    1. Prompt for and read the user's choice (rock (1), paper (2) or scissors(3)).
    2. Display the choice entered by the user to the screen.
    3. Generate a random number (1-3) which represents the computers choice; rock (1), paper (2), scissors (3).
    4. Display the choice generated by the computer to the screen.
    5. Determine the winner based on the rules described below.
    6. Display the winner to the screen.

    Use the systematic problem solving strategy presented in lecture slides in order to help you solve this problem.

    Rock-paper-scissors game play: The objective is to select a gesture which defeats the computer. Gestures are resolved as follows:
    - Rock crushes scissors – rock wins.
    - Paper covers rock – paper wins.
    - Scissors cut paper – scissors win.

    If both players choose the same gesture, the game is a draw.
    (Further information on the game can be found here: http://en.wikipedia.org/wiki/Rock-paper-scissors).

It is recommended that you develop the program in the following stages:

Stage 1:

Prompt for and read the user's choice (rock (1), paper (2) or scissors(3)).

```
Rock(1), Paper(2) or Scissors(3)?
```

The user will enter a number 1, 2, or 3. Use an if statement to display the choice entered by the user to the screen. i.e. if the user enters 3, your program should display the following text:

```
You chose scissors.
```

Stage 2:

Generate a random number (1-3) which represents the computers choice; rock (1), paper (2), scissors (3). Display the choice generated by the computer to the screen. i.e. if the random number generated was 2, your program should display the following text:

```
Computer chose paper.
```

Stage 3:

Determine the winner based on the rules described below.

- Rock crushes scissors – rock wins.
- Paper covers rock – paper wins.
- Scissors cut paper – scissors win.

If both players choose the same gesture, the game is a draw.

Stage 4:

Display the winner to the screen. i.e.

```
You win - scissors cut paper!
```

Stage 5:

Please make sure your program behaves as demonstrated in the sample output below. Also, please make sure that your output adheres to the sample output below (user's sample input appears in **bold**).

Sample output:

Example 1:

```
Rock(1), Paper(2) or Scissors(3)? 3
You chose scissors.
Computer chose paper.
You win - scissors cut paper!
```

Example 2:

```
Rock(1), Paper(2) or Scissors(3)? 1
You chose rock.
Computer chose rock.
Draw - no winner!
```

Example 3:

```
Rock(1), Paper(2) or Scissors(3)? 1
You chose rock.
Computer chose paper.
You lose - paper covers rock!
```

Question 1 (part b):
Include a while loop so that the user may play again if he/she wishes. Count how many games played and display this at the end of the session (once the user has finished playing).

Question 1 (part c):
Use a while loop (refer to lecture slides - using a while loop for error checking of user input) to make sure the user enters a valid number (i.e. 1, 2, or 3). If an invalid number is entered, the program should display an error message and re-prompt for a valid number.

Question 2
Modify the game of rock-paper-scissors that you implemented in question 1 to ensure that your solution defines and uses the functions described below:

a. Write a function called `display_details` to display your name, student id, etc. to the screen. This function accepts no parameters and does not return anything. Your function should produce the following output (with your details).

```
Author: your name
Email Id: your email id
This is my own work as defined by the
University's Academic Misconduct Policy.
```

b. Write a function called `get_choice` to prompt for, read and validate the user's choice. This function accepts no parameters and returns the user's choice.

Checkpoint: Please make sure your supervisor sees both your program output and code.

Question 3
Write a program that generates a seven-digit lottery number. The program should generate seven random numbers, each in the range of 0 through 9, and assign each number to a list element. Hint: Use a loop in order to assign each random number to a list element. Display the contents of the list (Hint: you will need to use another loop to do this). (Modified: Gaddis, Tony. Programming Exercises, Chapter 8).

Your output should be as follows:
```
5, 1, 4, 5, 9, 2, 3
```

Question 4

Write a program that prompts for and reads a string containing a person's full name (first, middle and last names) and then builds a new string containing their first, middle, and last initials. Display the new string to the screen. For example, if the user enters John William Smith, the program should display: J. W. S.
(Modified: Gaddis, Tony. Programming Exercises, Chapter 9).

Question 5

Write a program that asks the user to enter a series of single-digit numbers with nothing separating them. The program should display the sum of all the single digit numbers in the string. For example, if the user enters 2514, the program should display 12, which is the sum of 2, 5, 1 and 4. You may assume that the string does not contain non-digit characters.
(Modified: Gaddis, Tony. Programming Exercises, Chapter 9).

Question 6

Write a function called `read_colour` that prompts the user to enter a primary colour, either "red", "blue", or "yellow". Validate the colour entered by the user. If the user enters anything other than "red", "blue", or "yellow", the function should display an error message and continue to prompt until the user enters a 'valid' colour choice (i.e. . "red", "blue", or "yellow"). The function takes no parameters and returns the colour entered by the user. Defining the function does not execute the function – you will need to call the function from elsewhere in the program. Call function `read_colour` once it has been defined in order ensure that it is working correctly.

Question 7 (part a)

In each of the problems below, write (as a set of steps) how you would solve the problem, and then write the code. Do not write any code until you believe you can see how to solve the problem. Ignore issues such as the user entering bad data (i.e. text when they should be entering a number) – we will assume the user enters the correct data. **DO NOT** use a list to solve these problems. Your solutions **must** use a loop.

a. Write a short program that will read and sum numbers until the user enters a negative number.
b. Write a short program that will ask the user to enter a sequence of positive integer values (ending with zero or any negative value) and print out the number of values entered and the largest value entered.
c. Write a short program that will ask the user to enter a sequence of positive integer values (ending with zero or any negative value) and will print out the largest difference between the values entered.
Hint: the largest difference is the difference between the largest value entered and the smallest value entered. This is similar to the previous problem, except that you will also need to keep track of the smallest and the largest values entered.

Question 7 (part b)

In question 7 (part a), you were required to write a short program to read and sum numbers until the user enters a negative number. Modify your solution so that:

a. The program continues to loop until the user enters a negative number, or the sum of the numbers is greater than 15. Either of these conditions will cause the program to stop looping.
b. Display a message stating why the program has stopped looping – that is, because the user reached a total greater than 15, or a negative number was entered.
c. Display the total every time it is updated (i.e. within the loop).

Question 8

Given a list of menu items such as that seen below, write a program that creates a new list comprised of only the soups. Use this new list to randomly select and display a 'Soup of the day'.

```
menu = ['Lamb steak','Tomato soup','Fried rice','Pumpkin soup','Pea soup']
```

Question 9

Describe what the following program does. Check your answer using Python.

```
list1 = ['a',1,'b',2]
list2 = []

for element in list1:
    element = str(element)
    list2.append(element)

print(list1)
print(list2)
```

Question 10 (Prisoner Dilemma)

Two players compete in a simple game of choice. On each turn, both parties must decide whether to cooperate or betray one another. If they both cooperate (share), they each receive 2 coins. If they both betray (steal), nobody wins anything. But if one betrays and the other cooperates, the betrayer receives 3 coins, and their trusting opponent will lose 1.

Write a program to simulate this game. The user will compete against the computer, with the computer choosing to share or steal at random. On each turn, confirm each player's choice by displaying them to the screen; then evaluate and display the resulting outcome. Also display the current coin total of both user and computer at the end of each turn. The game will last for 5 turns, after which a winner is declared and the user can choose whether or not to start a new game.

Sample Output:

```
Please enter your choice, "share" or "steal": share
You chose:      share
Computer chose: steal

Computer gains 3 coins, You lose 1!

Your Score: -1
Comp Score: 3

Please enter your choice, "share" or "steal": share
You chose:      share
Computer chose: steal

Computer gains 3 coins, You lose 1!

Your Score: -2
Comp Score: 6

Please enter your choice, "share" or "steal": steal
You chose:      steal
Computer chose: steal

You both chose to steal! Nobody gets anything.
```

```
Your Score: -2
Comp Score: 6

Please enter your choice, "share" or "steal": steal
You chose:      steal
Computer chose: steal

You both chose to steal! Nobody gets anything.

Your Score: -2
Comp Score: 6

Please enter your choice, "share" or "steal": steal
You chose:      steal
Computer chose: share

You gain 3 coins, Computer loses 1!

Your Score: 1
Comp Score: 5

Computer Wins!


Would you like to play again? [y/n] n
```

Please make sure you save and keep all of your practical and assignment work. Please ask your supervisor if you are having difficulties doing so.

End of practical 6.