**Team Application Exercises (tAPP-2)**

`Instructions`: Work on `problem 1` on your own to apply the concepts you learned in the pre-reading material to solve problems. Then discuss your code with your team members and present one solution as a team. Follow the same process for all the problems. Swap your team solution with another team for feedback. `Problem 5`: Complete a function (extra problem).

**Problem 1**

The code below should repeat the user's message the number of times that the user wants. The code has 5 lines that are not in the correct order. Rearrange the code by entering the number that corresponds to each line of code (whatever you think should be the first line of code will have number 1 in front of it).

| | |
|---|---|
| | `n = int(input("How many times should it be reapted? "))` |
| | `for i in range(n):` |
| | `print(message)` |
| | `while message != "":` |
| | `message = input("Enter a message (blank to quit): ")` |

**Problem 2**

This code will determine whether or not a string entered by the user is a palindrome or not is not. The code has 11 lines that are not in the correct order. Rearrange the code by entering the number that corresponds to each line of code (whatever you think should be the first line of code will have number 1 in front of it – seen in the table).

| | |
|---|---|
| | `is_palindrome = False` |
| | `if is_palindrome:` |
| | `else:` |
| | `i = 0` |
| | `if line[i] != line[len(line) - i - 1]:` |
| 1 | `line = input("Enter a string: ")` |
| | `print(line, "is not a palindrome")` |
| | `i = i + 1` |
| | `is_palindrome = True` |
| | `while i < len(line) / 2 and is_palindrome:` |
| | `print(line, "is a palindrome")` |

**Problem 3**

When the following code runs, it asks for the name and then the age. If the age is under or equal to 10, the message is Hi followed by the name.

The code has 13 lines that are not in the correct order. Rearrange the code by entering the number that corresponds to each line of code (whatever you think should be the first line of code will have number 1 in front of it – seen in the table).

| | |
|---|---|
| | `def main():` |
| | `if age <= 10:` |
| | `return data_tuple` |
| | `print("Hi", username)` |
| | `username = input("Enter your user name: ")` |
| | `message(username, age)` |
| 1 | `def get_data():` |
| | `def message (username, age):` |
| | `age = int(input("Enter your age: "))` |
| | `data_tuple = (username, age)` |
| | `username, age = get_data()` |
| | `print("Hello", username)` |
| | `else:` |

**Problem 4: Debugging code**

The code below should display a menu, add or subtract two numbers and display the corresponding message. However, the code is not working. Identify the issues that affect this code.

Identify the lines with errors. How many errors did you find and what are the errors?

Code with Errors

```
import random

def addition():
    num1 = random.randint(5,20)
    num2 = random.randit(5,20)
    print(num1, "+", num2, "= ")
    user_answer = int(input("Your answer: "))
    actual_answer = num1 + num2
    answers = (user_answer, actual_answer)
    return answer

def subtraction():
    num3 = random.randint(25,50)
    num4 = random.random(1,25)
    print(num3, "-", num4, "= ")
    user_answer = int(input("Your answer: "))
    actual_answer == num3 - num4
    answers = (user_answer, actual_answer):
    return answers

def check_answer(user_answer,actual_answer):
  if user_answer == actual_answer:
        print("Correct")
    else:
        print("Incorrect, the answer is ", actual_answer)

def main():
    print("1) Addition")
    print("2) Subtraction")
    selection = int(input("Enter 1 or 2: "))
    if selection = 1:
        user_answers, actual_answer = addition()
        check_answer(user_answer, actual_answer)
    elif selection == 2:
        user_answer, actual_answer = subtraction()
        check_answer(user_answer actual_answer)
  else:
        print("Incorrect selection")

main()
```

**Problem 5:**

This programme will ask the user to pick a low and high number, and then generate a random number between those two values and store it in a variable called “comp_num”. It will give the instruction “I am thinking of a number …” and then ask the user to guess the number they are thinking of.

**`Incomplete function:`** The third function should check to see if the `comp_num` is the same as the user’s guess. If it is, it should display the message “Correct, you win”, otherwise it should keep looking, telling the user if they are too low or too high and asking them to guess again (using a variable called `try_again`) until they guess correctly.

**Task:** Using a pen and paper, complete the function (with correct indentation).

```
import random

def pick_number():
    low = int(input("Enter the bottom of the range: "))
    high = int(input("Enter the top of the range: "))
    comp_num = random.randint(low, high)
    return comp_num

def first_guess():
    print("I am thinking of a number ... ")
    guess = int(input("What am I thinking of: "))
    return guess

def check_answer(comp_num, guess):









def main():
    comp_num = pick_number()
    guess = first_guess()
    check_answer(comp_num, guess)


main()
```