MA1008 Introduction to Computational Thinking Quiz 1
Answer all the nine questions in the spaces provided
AY 2022/2023, Semester 1, Week 5

Your Name: ___________________________________ Group: _____

Solutions

A question may have multiple correct answers. If the solution a student offers is different, please check the correctness before awarding the marks.

1. Give two reasons why there is a need to differentiate the data type of a variable in a programming language. (6 marks)

   Students can provide any two of three: (1) The underlying representations for different data types are different. (2) Different data types support different operations, and for these operations to work, we need to supply them with variables of the correct types. (3) The objects we wish to represent in a computer program are of different types in nature and require different data types.

2. Write down (a) the value and (b) the data type of the variable on the left-hand side of the following statements. (4 marks each)

   i. `x = 2 + 3 ** (12 / 4) // 9 * 2` 8.0, float

   ii. `y = 4 * 5 // 4 + - 6 % 7 % 6` 6, int

   iii. `z = 10 + 100 // 20 - 200 / (100 * 10 // 200 + 20) / 2` 11.0, float

3. Given the following variables: `P = True, Q = False, R = 1, S = 5, T = 10`

   a. Write down the result of each of the following Boolean expressions.
   b. Show the order of your evaluation by placing brackets around the operations, with one pair of brackets for each operator. (4 marks each)

   i. `((P and Q) or (((not P) and (not Q)) and (P or (not Q))))` False

   ii. `(((((R + S) > (T / S)) > (T - (2 * S))) and (T - (2 * R)))` 8 or True

   We can accept 8 or True as the solution. Python delivers 8, but this is a Boolean, so True OK.

   iii `(((((2 * T) == ((3 * S) + (4 * R))) and P) or Q)` False

4. The following program contains errors. Circle the errors and pen in the correction. (10 marks)

```
num1 = int(input('Key in a positive integer: ') ____________________

If num1 < 0 ________________________________________________________

num2 = -Num1 _______________________________________________________

print('You keyed in' num1, 'did you mean' num2?)   _________________

else _______________________________________________________________

print('You have keyed in', 'num1')  ________________________________
```

```
num1 = int(input('Key in a positive integer: '))
if num1 < 0:
   num2 = -num1
   print('You keyed in', num1, 'did you mean', num2, '?')
else:
   print('You have keyed in', num1)
```

5. Given

```
s1 = "Introduction to"
s2 = "Computational"
s3 = "Thinking"
```

i. What are printed in the following statements? (3 marks)

```
print(s1)                          Introduction to

print(s2)                            Computational

print(s3, 2022)                      Thinking 2022
```

ii. Keep the three print statements, but modify them such that they print the three strings in one line, with a space separating the words. (5 marks)

```
print(s1, end = " ")
print(s2, end = " ")
print(s3, 2022)
```

iii. Provide one print statement that prints the three strings in three separate lines. (5 marks)

```
print(s1, "\n", s2, "\n", s3, 2022)
```

6. You are to write a program to manage money donation from the public, report the sum collected and determine when to stop collecting. (a) State whether it is better to use a `for` loop or a `while` loop to control each of the following actions. (b) Explain your answers.

   i. Stop collecting when the sum reaches a pre-set target. (5 marks)

   while loop. We do not know how many times we have to collect to reach the target, so can only set the condition sum < target as the sentinel in the while loop.

   ii. Collect and report the sum from a pre-set number of donors. (5 marks)

   for loop. Since we know the number of people we need to collect from, it is more natural to set the range for that number in a for loop.

   iii. Stop collecting and report the sum when either a pre-set target amount or a given pre-set number of donors is reached. (5 marks)

   While loop. The two conditions are such that we cannot set a range for a for loop. So we need to use a while loop.

7. Write a Python program that repeatedly reads in floating point numbers and sums those satisfying the following rules:
   a. The number lies between -1000 and +1000, inclusive.
   b. The integer part of the number is divisible by 6 but not divisible by 5.
   c. Any number between -500 to +500, inclusive, is doubled before being added to the sum.

   Exit the reading when the number is less than -10000 or greater than 10000. Print the sum upon exiting from reading. You may assume that only numerical values are read in. (12 marks)

```
sum = 0
while True:
   num = float(input("Enter a floating point number: "))
   if num < -10000 or num > 10000:
      break
   elif -1000 <= num <= 1000:
      if int(num)%6 == 0 and int(num)%5 != 0:
         if -500 <= num <= 500:
            num *= 2
         sum += num

print(sum)
```

8. i. What are printed in the following Python code? (4 marks)

```
for counter in range(4, 20, 5):

    while True:

        print(counter)

        break

    else:

        continue
```

```
4
9
14
19
```

ii. Simplify the above code and yet produce the same result. (6 marks)

```
for counter in range(4, 20, 5):
    print(counter)
```

9. The following Python program reads in two integers `A` and `B` and prints `True` if all the digits of `B` are in `A` and prints `False` otherwise. The variable `inA` carries the eventual result. Examples: if `A` = `123` and `B` = `13`, `inA` is `True`; `A` = `123` and `B` = `32321`, `inA` is `True`; `A` = `123` and `B` = `324`, `inA` is `False`. Fill in the blanks to complete the program. (10 marks)

```
A = int(input("Enter integer A: "))
B = int(input("Enter integer B: "))

inA = True    # initialise inA

for c in str(B):                  # pick out each digit in B

    if c not in str(A):           # if this digit is not in A

        inA = False               # reset value of inA

        break                     # exit the loop

print (inA)    # print the outcome
```

<> <> <> THE END <> <> <>