



## COMP 1039 Problem Solving and Programming

### Practical 5

#### Debugging

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, date, brief program description and the University's academic misconduct statement.

#### Question 1

Write a program that will prompt for and read a number (positive integer) and calculate the hailstone sequence for the number ([http://en.wikipedia.org/wiki/Collatz\\_conjecture](http://en.wikipedia.org/wiki/Collatz_conjecture)). Display each step taken and the number of steps it took to reach 1 (see sample output below). Make sure you validate the input, that is, only allow positive integers to be input by the user.

The algorithm for the hailstone sequence can be described as follows:

Pick a positive integer and call it  $n$

If  $n$  is even, divide it by two ( $n/2$ )

If  $n$  is odd, multiply it by three and add one ( $3n + 1$ )

Continue this process until  $n$  is equal to one.

#### Sample output:

```
Please enter a number: 15

15 is odd, so I calculate  $3n + 1$ : 46
46 is even, so I divide by 2: 23
23 is odd, so I calculate  $3n + 1$ : 70
70 is even, so I divide by 2: 35
35 is odd, so I calculate  $3n + 1$ : 106
106 is even, so I divide by 2: 53
53 is odd, so I calculate  $3n + 1$ : 160
160 is even, so I divide by 2: 80
80 is even, so I divide by 2: 40
40 is even, so I divide by 2: 20
20 is even, so I divide by 2: 10
10 is even, so I divide by 2: 5
5 is odd, so I calculate  $3n + 1$ : 16
16 is even, so I divide by 2: 8
8 is even, so I divide by 2: 4
4 is even, so I divide by 2: 2
2 is even, so I divide by 2: 1
```

The hailstone sequence took 17 steps to reach 1

## Question 2

This question builds on an exercise undertaken in an earlier practical - please complete the exercise (seen below) if you haven't already done so (you may also like to refer to and use the solution provided on the course website):

*Create a file in order to complete this exercise. Many games require a player to roll two dice. The number on each die can vary from 1 to 6.*

- a. *Use the `random.randint()` function to create a simulation of one roll of one die. Store the result in a variable called `die1`. (hint: remember to place `import random` at the top of your file so that you may make use of the `randint()` function).*
- b. *Repeat part a. to create a simulation of the value rolled with a second die. Store the result in a variable called `die2`.*
- c. *Display the results of the die to the screen in the following format:*

*If the values of the die are the same – display the message you rolled a pair of... i.e.:*

```
You rolled a pair of 2s!
```

*Otherwise, display the values of the die. i.e.:*

```
You rolled a 2 and a 6
```

**Now let's modify the program as follows:**

1. Add a while loop that repeats the steps above (a, b & c) 10 times.
2. Count how many times a pair is rolled.
3. Display the results to the screen - refer to sample output below.

**Now let's modify the program again as follows:**

1. Use a while loop that repeats the steps above (a, b & c) 10 times instead of a for loop.

Your output should be presented as follows:

### Sample output 1:

```
You rolled a 6 and a 5
You rolled a 1 and a 5
You rolled a 4 and a 5
You rolled a 3 and a 5
You rolled a 4 and a 6
You rolled a 2 and a 5
You rolled a 3 and a 4
You rolled a 2 and a 1
You rolled a 3 and a 6
You rolled a 5 and a 6
```

```
You didn't roll any pairs - so sad :(
```

**Sample output 2:**

```
You rolled a 2 and a 5
You rolled a 3 and a 4
You rolled a 6 and a 4
You rolled a 6 and a 5
You rolled a 5 and a 3
You rolled a 1 and a 2
You rolled a 5 and a 6
You rolled a pair of 6s!
You rolled a 3 and a 2
You rolled a 5 and a 4
```

You only rolled one pair.

**Sample output 3:**

```
You rolled a 3 and a 6
You rolled a pair of 5s!
You rolled a 5 and a 4
You rolled a 4 and a 2
You rolled a 3 and a 2
You rolled a pair of 3s!
You rolled a 4 and a 6
You rolled a 3 and a 1
You rolled a pair of 5s!
You rolled a 1 and a 4
```

You rolled a pair 3 times!

**Question 3**

The colours red, blue and yellow are known as the primary colours because they cannot be made by mixing other colours. When you mix two primary colours, you get a secondary colour, as shown here:

- When you mix red and blue, you get purple.
- When you mix red and yellow, you get orange.
- When you mix blue and yellow, you get green.

Write a program that prompts the user to enter the names of two primary colours to mix. If the user enters anything other than "red", "blue", or "yellow", the program should display an error message. Otherwise, the program should display the name of the secondary colour that results.

(Modified: Gaddis, Tony. Programming Exercises, Chapter 4).

#### Question 4

Write a program to "sing" the '99 Bottles of Beers' song ([https://en.wikipedia.org/wiki/99 Bottles of Beer](https://en.wikipedia.org/wiki/99_Bottles_of_Beer)). You **must** use loops in your solution. You **must** also validate all user input. Your code should display the following output to the screen:

##### Sample output 1:

```
Would you like to sing a song [y|n]? y

How many verses of the song do you wish to sing? 2
100 bottles of beer on the wall
100 bottles of beer
If one of those bottles should happen to fall
99 bottles of beer on the wall

99 bottles of beer on the wall
99 bottles of beer
If one of those bottles should happen to fall
98 bottles of beer on the wall

That was fun! Would you like to sing again [y|n]? n

Thanks for singing along!  : )
```

##### Sample output 2:

```
Would you like to sing a song [y|n]? p
Would you like to sing a song [y|n]? z
Would you like to sing a song [y|n]? y

How many verses of the song do you wish to sing? 10000
Not possible my friend ...

How many verses of the song do you wish to sing? 0
Not possible my friend ...

How many verses of the song do you wish to sing? 4
100 bottles of beer on the wall
100 bottles of beer
If one of those bottles should happen to fall
99 bottles of beer on the wall

99 bottles of beer on the wall
99 bottles of beer
If one of those bottles should happen to fall
98 bottles of beer on the wall

98 bottles of beer on the wall
98 bottles of beer
If one of those bottles should happen to fall
97 bottles of beer on the wall

97 bottles of beer on the wall
97 bottles of beer
If one of those bottles should happen to fall
96 bottles of beer on the wall
```

That was fun! Would you like to sing again [y|n]? y

How many verses of the song do you wish to sing? 1  
100 bottles of beer on the wall  
100 bottles of beer  
If one of those bottles should happen to fall  
99 bottles of beer on the wall

That was fun! Would you like to sing again [y|n]? n

Thanks for singing along! : )

### Sample output 3:

Would you like to sing a song [y|n]? y

How many verses of the song do you wish to sing? 100  
100 bottles of beer on the wall  
100 bottles of beer  
If one of those bottles should happen to fall  
99 bottles of beer on the wall

99 bottles of beer on the wall  
99 bottles of beer  
If one of those bottles should happen to fall  
98 bottles of beer on the wall

98 bottles of beer on the wall  
98 bottles of beer  
If one of those bottles should happen to fall  
97 bottles of beer on the wall

97 bottles of beer on the wall  
97 bottles of beer  
If one of those bottles should happen to fall  
96 bottles of beer on the wall

96 bottles of beer on the wall  
96 bottles of beer  
If one of those bottles should happen to fall  
95 bottles of beer on the wall

:  
: *...you get the idea (removed to save paper)... :*  
:

3 bottles of beer on the wall  
3 bottles of beer  
If one of those bottles should happen to fall  
2 bottles of beer on the wall

2 bottles of beer on the wall  
2 bottles of beer  
If one of those bottles should happen to fall  
1 bottle of beer on the wall

1 bottle of beer on the wall  
1 bottle of beer  
If one of those bottles should happen to fall

```
No bottles of beer on the wall!!
```

```
That was fun! Would you like to sing again [y|n]? n
```

```
Thanks for singing along! : )
```

**Checkpoint:** Please make sure your supervisor has seen the work you have completed so far.

### Question 5

Write a program that evaluates and displays all of the classic children's multiplication tables between 1 and 12 (inclusive).

### Question 6

Describe what the following program does. You may check your answer using Python.

```
index1 = 0
index2 = 0

input1 = int(input("Enter an integer: "))
input2 = int(input("Enter another integer: "))

while index1 < input1:
    while index2 < input2:
        print("#",end="")
        index2 = index2 + 1
    print()
    index2 = 0
    index1 = index1 + 1
```

### Question 7

Given the following menu, write code that will validate the user's input. That is, write code for the Pizza Menu program to restrict the user input to only options on the menu (1, 2, or 3).

```
print("Pizza Menu:")
print("=====")
print("1. Margherita")
print("2. Hawaiian")
print("3. Pepperoni")

user_input = input("\nPlease make a selection: ")
```

## PART II: Debugging...

Use the debugging techniques described in the week 5 video and slides in order to identify and fix the bugs in the following code. In particular, **start by using one of the most powerful debugging tools... the humble print statement. Please ensure that you also run the IDLE debugger over the code.** The following code (exercises 1, 2 & 3) is available from the course website (this will save you from typing it in).

### Debugging Exercise 1

Find and fix the bug in the following code:

```
count = 1

while count <= 5:
    count = count - 1

print('Count is', count)
```

The above code should display the follow output to the screen:

```
Count is 6
```

### Debugging Exercise 2

Find and fix the bug in the following code:

```
mark = 69
grade = ''

if mark >= 85:
    grade = 'HD'
elif mark >= 75:
    grade = 'D'
elif mark >= 65:
    grade == 'C'
elif mark >= 55:
    grade = 'P1'
elif mark >= 50:
    grade = 'P2'
elif mark >= 40:
    grade = 'F1'
else:
    grade = 'F2'

print('The grade is:', grade)
```

The above code should display the follow output to the screen:

```
The grade is: C
```

### Debugging Exercise 3

Find and fix the bug in the following code:

```
import random

play = 'y'
while play == 'y':

    # generate random number 1 - 100 inclusive
    number = random.randint(1,100)

    # prompt for and read user's guess
    guess = int(input('Please enter your guess: '))
    print('')

    # determine whether user guessed correct random number
    while guess == number:
        if guess < number:
            print('Too low - please try again!')
        else:
            print('Too high - please try again!')

        # prompt for and read user's guess
        guess = int(input('Please enter your guess: '))
        print('')

    print('Well done - you guessed it!')

    # prompt for and read whether the user would like to play again
    play = input('Would you like to play again (y/n)? ')
    print('')
```



The above code should display the follow output to the screen:

Please enter your guess: 15

Too low - please try again!

Please enter your guess: 75

Too high - please try again!

Please enter your guess: 46

Well done - you guessed it!

Would you like to play again (y/n)? y

Please enter your guess: 23

Too low - please try again!

Please enter your guess: 88

Too high - please try again!

Please enter your guess: 54

Too low - please try again!

Please enter your guess: 73

Well done - you guessed it!

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 5.***