COMPSCI 101
Principles of Programming
Revision Exercises
COMPSCI 101 Test – General Information

• Worth 15% of your final mark.

• 75 minutes.

• Closed book, no calculators.
COMPSCI 101 Test – Question Breakdown

- 5 multi-part questions each worth 20 marks:
  - 2 from my part
  - 3 from Jing’s part

- Questions will involve:
  - Writing code to perform a given task
  - Figuring out the output of provided code
Lectures 1 – 13

1. Introduction
2. Variables, program execution, doing calculations, print()
3. Expressions, mathematical operators, the math module
4. Strings, objects, the len() function, string slices
5. Manipulating strings, string functions, dot notation
6. Getting user input, converting between types, generating random numbers
7. Defining functions 1
8. Defining functions 2
9. Code tracing, divide a program into separate tasks
10. Boolean expressions, if statements, equality and floats
11. if ... else, if ... elif statements, nested ifs
12. while loops
13. The Python range() function, for ... in loops

Labs 1 to 4
Go through lecture exercises.

Go through lab exercises.

Do questions from past year tests and exams.
- Can be found on the COMPSCI 111 website – Tests and Exams page
  
  https://www.cs.auckland.ac.nz/courses/compsci101s1c/exams/

Don’t memorize lecture slides or exercises.
- Focus on the problem being solved
- Exercises involving similar problems will have similar solutions!
Arrive to the test early. Avoid unnecessary stress rushing.

Use the 5 minute reading time to your advantage.

Start with the easy problems.

If you can’t solve a problem entirely, write code dealing with the parts of the problem you can address.
  - Partial marks are allocated so don’t leave anything blank!

If you have time check your work before leaving!
KEEP CALM AND GOOD LUCK
Assume that the Python variables dog, cat and rabbit have all been assigned integer values. Which one of the following would best describe the outcome of the following piece of code?

```
rabbit = dog
cat = rabbit
dog = cat
```

(a) Each variable would store the same value (the initial value of rabbit).
(b) The values in variables rabbit and dog would be swapped.
(c) Each variable would store the same value (the initial value of dog).
(d) The values in variables cat and dog would be swapped.
(e) The values in variables rabbit and cat would be swapped.
Complete the output of the following code:

```python
var1 = 59
var2 = 71
var3 = var1
var1 = var2
var2 = var3
print("Var1: ", var1, "\nVar2: ", var2, sep=" ")
```

Output:

```
Var1: 71
Var2: 71
```
What is the output of the following code?

```
result = 1 + 2 ** 3 / 3 // 2 % 5
print(result)
```

What is the output produced by the following code?

```
value = 3 ** (1 + 1 * 2) // 5 % (20 / 2)
print(value)
```
What is the output of the following code?

```python
import random
var1 = round(57.789, 1)
var2 = abs(3 - 60)
var3 = random.randrange(1, 59, 2)
print(max(var1, var2, var3))
```

Which of the following could NOT be produced by the code below?

```python
import random
var1 = random.randrange(19, 41, 4)
var2 = random.randrange(17, 45, 5)
var3 = random.randrange(42, 18, -3)
print(max(var1, var2, var3))
```

(a) 21
(b) 17
(c) 39
(d) 32
(e) 42
The number $e$ is an important mathematical constant that is the base of the natural logarithm. It is equal to:

$$2.7182818284...$$

The following is an approximation of $e$ accurate to 5 decimal places:

$$e = \frac{1}{\left(1 - \frac{1}{10^6}\right)^{10^6}}$$

Write a program to implement this approximation. Your program should produce the following output that has been rounded to 5 decimal places:

The mathematical constant "e" to five decimal places = 2.71828
"""
Author: COMPSCI 101 Class
Program calculates an approximation of e accurate to 5 decimal places
"""
Revision Exercises – User Input/String Manipulation

What is the output of the following code?

```python
sentence = "That kazoo sounds beastly!"
substring = sentence[0] + sentence[8:11] + sentence[-7:-4] + sentence[-2:]
print(substring)
```

What is the output produced by the following code if the user enters "nothing is impossible" at the prompt?

```python
string = input("Please enter a string: ")
left_space_index = string.find(" ")
right_space_index = string.rfind(" ")
if left_space_index != right_space_index:
    string = string[right_space_index + 1:] + " " + string[left_space_index + 1:right_space_index] + " " + string[:left_space_index]
print(string)
```
A file extension enables an operating system like Windows to determine the type of file it is dealing with. The file extension consists of one or more characters following the last period ("." ) in a file name.

Write a program that asks the user to input a file name. Your program should identify the file extension and print out a message informing the user what it is. Some example output:

Please enter the name of the file: Tutorial.py
The file Tutorial.py has an extension of py

Please enter the name of the file: Trick.pdf.avi.docx
The file Trick.pdf.avi.docx has an extension of docx
Revision Exercise – User Input/String Manipulation

"""
Author: COMPSCI 101 Class
Program extracts and prints the file extension from a provided file name
"""
Revision Exercises – User Input/String Manipulation

Write a program so that it asks the user to enter a string. The program will then create a new string, where one character in the original string, selected at random, is replaced by a random letter from the alphabet. The function will then print the new string. The example below shows a possible output produced by the completed program if the user enters the string “Videogames”

Please enter a string: Videogames
Your new string is: Videogases

You can assume that the user will always enter a word at least 1 character long.
Revision Exercises – User Input/String Manipulation

""
Author: COMPSCI 101 Class
Program replaces a character in a string with a random letter from the alphabet
""

import random
alphabet = "abcdefghijklmnopqrstuvwxyz"
Using the code trace technique taught in lectures, perform a code trace on the following program and show the output.

```python
def first(num1):
    num2 = 3
    print("1.", num1)
    return second(num1 + num2) - 3

def second(num1):
    print("2.", num1)
    return num1 * 2

def main():
    num1 = 5
    num2 = first(num1)
    print("3.", num2)
    num2 = second(num2 - num1)
    print("4.", num2)

main()
```

Output: 

1. 5

2. 5

3. 2

4. 4
What is the output of the following code?

def function1(num1, num2, num3):
    total = num1 + num2 + num3
    total = total - max(num1, num2, num3)
    result = total - min(num1, num2, num3)
    return result

def main():
    print("1.", function1(58, 39, 86))
    print("2.", function1(120, 198, 320))
    print("3.", function1(60, 20, 50))

main()
Complete the make_banner() function below which takes a single string parameter that provides the information needed to print a banner. The string is formatted as follows:

- the last character of the string is used for the banner frame.
- the second to last character of the string determines how wide the banner frame is on either side of the banner text. You can assume this will be a character between 0 and 9.
- the remainder of the string is the banner text. You can assume that the banner text will be at least one character long.

For example, when the following program is executed with the completed function, the output is:

```
######################
###COMPSCI 101 Exam###
######################
```
def main():
    make_banner("COMPSCI 101 Exam3#")
main()
Write the following boolean expressions. Assume that the variable, `value`, has been initialized.

- `value` is less than 80 but greater than or equal to 75

- `value` is not equal to 100 or 0

- `value` between 50 and 60 (inclusive) or equal to 75
What is the output of the following code?
def function3(x, y, z):
    if x == 5 or y > 3:
        if x > 4 and z == 6:
            print("I")
        else:
            if y == 6 and z >= x:
                print("II")
            else:
                print("III")
    else:
        print("IV")
def main():
    function3(2, 4, 6)
main()

What is the output of the following code?
def mystery(num1, num2):
    if (num1 >= 50 and num2 < 40):
        print('case 1')
    if (num1 > 50 or num2 < 30):
        print('case 2')
    else:
        print('case 3')
def main():
    mystery(50, 30)
    mystery(50, 20)
main()
Revision Exercises – If/elif/else statements

Complete the following function `get_grade()` which converts an integer mark into its corresponding grade. The grade boundaries to be used in this program are as follows.

- A – greater than or equal to 90;
- B – less than 90 but greater than or equal to 80;
- C – less than 80 but greater than or equal to 70;
- D – less than 70 but greater than or equal to 60;
- F – less than 60.

For instance, the following program with the completed function gives the output:

89 – B grade
65 – D grade
def get_grade(mark):
What is the output of this code?

```python
term = 0
step = 1
end = 30
while term < end:
    print(term)
    term = term + step
    step = step + 2
print("term:", term)
print("step:", step)
```

What is the output of this code?

```python
previous = 3
number = 6
for num in range(number, 12, 3):
    print(previous, num)
    previous = num
print("end:", previous)
```
Complete the `print_upside_triangle()` function below which takes an integer as a parameter and prints an upside down triangle of the size specified in the parameter. For example, the following code:

```python
print_upside_triangle(4)
```

will produce the output:

```
####
###
##
#
```

Revision Exercises – Loops
<table>
<thead>
<tr>
<th>Using a while loop</th>
<th>Using a for loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>def print_upside_triangle(size):</td>
<td>def print_upside_triangle(size):</td>
</tr>
</tbody>
</table>
Write a program that lets the user play a simple guessing game. The program first generates a random number between 1 and 100 (inclusive). The game then prompts the user to guess the random number. If the user guesses too high or too low, the program informs them of this. The game ends when the user guesses the random number. An example of the program running is shown below:

Guess a number between 1 and 100

Guess the number: 50
Too low, try again.
Guess the number: 75
Too low, try again.
Guess the number: 87
Too high, try again.
Guess the number: 80
Too low, try again.
Guess the number: 83
Too low, try again.
Guess the number: 85
Well done!
See you later.