# THE UNIVERSITY OF AUCKLAND

**Semester Two, 2017**
**Campus: City**

## TEST

## COMPUTER SCIENCE
## SOLUTIONS
**Principles of Programming**

**(Time Allowed: 75 Minutes)**

**NOTE:**
You must answer **all** questions in this test.
**No** calculators or smart watches are permitted.
Answer in the space provided in this booklet.
There is space at the back for answers which overflow the allotted space.

| | |
|---|---|
| **Surname** | |
| **Forenames** | |
| **Preferred Name (if different to forenames)** | |
| **Student ID** | |
| **Username** | |
| **Lab Time** | |

| | |
|---|---|
| **Q1** (/20) | **Q4** (/20) |
| **Q2** (/20) | **Q5** (/20) |
| **Q3** (/20) | **TOTAL** (/100) |

## Question 1 (20 marks)

a) Complete the output produced by the following code.

```
num1 = 5
num2 = num1
num1 = num1 + 4
num3 = num2
num2 = num2 + 2

print("Three numbers:", num1, "-", num2, "-", num3)
```

```
Three numbers: 9 - 7 - 5
```

(2 marks)

b) Complete the output produced by the following code.

```
result = 5 * 3 ** 2 + 3 - 6 // 2 ** 2
print("Result:", result)
```

```
Result: 47
```

(2 marks)

c) Complete the output produced by the following code.

```
result1 = 15 % 4 + 6 % 8
result2 = 11 // 2 + 6 / 3
print("Results:", result1, '-', result2)
```

```
Results: 9 - 7.0
```

(2 marks)

d) Given the following code, give the **type** of the three objects: `object1`, `object2` and `object3`.

```
object1 = "765"
object2 = len(object1 * 2) / 3
object3 = object1[-2] == 6
```

```
Type of object1: str

Type of object2: float

Type of object3: bool
```

(2 marks)

e) Assume the variable, `value`, has been initialised. Write the boolean expression which tests if `value` is a multiple of 4 between 20 and 350 (both not inclusive)..

```
value % 4 == 0 and value > 20 and
                          value < 350
```

(2 marks)

f) Complete the output produced by the following code.

```
result1 = True
result2 = False
final_result = result1 and result2 or result1 and not result2
print("Final result:", final_result)
```

```
Final result: True
```

(2 marks)

g) Complete the output produced by the following code.

```
words = "damaged goods should be called bads"
result = words[2: 5] + words[-3:] + words[9]
print("Result:", result)
```

```
Result: magadso
```

(2 marks)

h) Complete the output produced by the following code.

```
words = "over the top"
index1 = words.find(' ')
index2 = words.rfind(' ')
word1 = words[:index1]
word2 = words[index2:]
combined = word1 + word2
print("Combined:", combined)
```

```
Combined: over top
```

(2 marks)

i) Complete the output produced when the following `main()` function is executed.

```
def get_result(num1, num2):
    new_num = num1 * num2
    digits = str(new_num)
    new_num = digits[0] + digits[-1]
    return int(new_num)

def main():
    result = get_result(4, 2)
    print("Result:", result)
```

```
Result: 88
```

(2 marks)

j) Give the smallest and the largest number which could be assigned to the variable, `result`. Assume that the `random` module has been imported.

```
result = random.randrange(3, 23, 5)
```

```
Smallest: 3   largest: 18
```

(2 marks)

## Question 2 (20 marks)

a) Give the output produced when the following `main()` function is executed.

```
def function_ifs(a, b, c):
    if a > b and a > c:
        print("A")
    elif not a > 5 or a > c:
        print("B")
    else:
        print("C")

    if a > b and c > b:
        print("D")
    elif b > c or c > 5:
        print("E")
        if not a < 20:
            print("F")

    print("G")

def main():
    function_ifs(10, 11, 8)
```

```
B
E
G
```

(6 marks)

b) Add code to the `main()` function on the next page so the program does the following:

1. Assigns 5451 to a variable named `balance`. (Already done.)
2. Assigns 3456 to a variable named `amount`. (Already done.)
3. Adds 738 to the variable, `balance`.
4. Divides the variable, `amount`, by 29 and ignores any remainder. Subtracts this result from the variable, `balance`.
5. Performs a modulus 123 on the variable, `balance`.

```
def main():
```

```
balance = 5451
amount = 3456

balance = balance + 738
balance = balance - amount // 29
balance = balance % 123
```

(9 marks)

c) The following `main()` function should execute as shown in the two example screenshots below (the user input is shown in a larger bold font):

```
Enter mark out of 25: 20
Final result: 80% - PASS
```

```
Enter mark out of 25: 10
Final result: 40% - NOT A PASS
```

The program contains two errors which means that the program does not execute correctly. Circle the two statements which contain an error and write the corrected statements in the space provided.

```
def main():
    prompt = "Enter mark out of 25: "
    mark25 = input(prompt)
    mark100 = round(mark25 * 4)

    pass_message = "PASS"
    if mark100 < 50:
        pass_message = "NOT A PASS"

    percent_str = mark100 + '%'
    print("Final result:", percent_str, "-", pass_message )
```

Correction 1:

```
mark25 = int(input(prompt))
```

Correction 2:

```
percent_str = str(mark100) + '%'
```

(5 marks)

## Question 3 (20 marks)

a) Complete the output produced when the following `main()` function is executed.

```
def get_code_str(digits):
    code_numbers = "9067584312"
    code_letters = "ABCDEFGHIJ"

    index1 = code_numbers.find(digits[0])
    index2 = code_numbers.find(digits[1])
    code = code_letters[index1] + code_letters[index2]
    return code

def main():
    code = get_code_str("86")
    print("Code:", code)
```

```
Code: FC
```

(6 marks)

b) Complete the `display_starry_name()` function which has one string parameter, `name`. The function prints a single star followed by the middle part of the parameter string, `name`, all in lowercase letters followed by another star. You can assume that the parameter string always contains at least two characters. For example, executing the following program with the completed function, prints:

```
*imo*
*ennife*
**
*ebastia*
```

```
def main():
    name = "simon"
    display_starry_name(name)
    display_starry_name("JENnifer")
    display_starry_name("LI")
    display_starry_name("SebASTian")
```

```
def display_starry_name(name):
```

```
name = name.lower()
middle_part = name[1:len(name) - 1]
starry_name = '*' + middle_part + '*'
print(starry_name)
```

(8 marks)

c) In the `main()` function below, complete the call to the `get_cost()` function so that the output of the program is:

```
Cost: $11.5
```

```
def get_cost(number_of_fillings, with_sauce, with_cheese):
    cost = number_of_fillings * 2
    if with_sauce:
        if with_cheese:
            cost = cost + 2.5
        else:
            cost = cost + 1
    else:
        if with_cheese:
            cost = cost + 1.5
        else:
            cost = cost - 1
    return cost

def main():
```

```
cost = get_cost(5, False, True)
```

(6 marks)

```
    print("Cost: $", cost, sep = "")

main()
```

## Question 4 (20 marks)

a) Give the output produced by the following code.

```
number = 0
num2 = 23
counter = 7

while counter > 3:
    number = number + 1
    num2 = num2 - 5
    print(number, '-', num2)
    counter = counter - 2

print(number, '-', num2, counter)
```

```
1 - 18
2 - 13
2 - 13 3
```

(10 marks)

b) The following program is a guessing game where the user is prompted to guess a computer generated random number between 1 and 9 inclusive. The user is prompted to guess the number until the number they enter is greater than or equal to the computer number. The program also keeps count of the number of guesses entered by the user. In the space provided in the `guess_number()` function add a `while` loop which:

- prompts the user for a number until the number entered by the user is greater than or equal to the computer number,

and,

- keeps track of the number of guesses entered by the user.

Two example executions of the completed program are shown in the textboxes on the next page.

```
Enter your guess (1 to 9 inclusive): 3
Enter your guess (1 to 9 inclusive): 5
Enter your guess (1 to 9 inclusive): 7
Enter your guess (1 to 9 inclusive): 8

Computer number: 8
Number of guesses: 4
Well Done!
```

```
Enter your guess (1 to 9 inclusive): 3
Enter your guess (1 to 9 inclusive): 5

Computer number: 4
Number of guesses: 2
5 is too big!
```

```
import random

def main():
    number = random.randrange(1, 10)
    guess_number(number)

def guess_number(computer_num):
    prompt = "Enter your guess (1 to 9 inclusive): "
    guesses = 1
    user_num = int(input(prompt))
```

```
while user_num < computer_num:
    user_num = int(input(prompt))
    guesses = guesses + 1
```

(10 marks)

```
    print()
    print("Computer number:", computer_num)
    print("Number of guesses:", guesses)
    if user_num > computer_num:
        print(user_num, "is too big!")
    elif guesses < computer_num:
        print("Well Done!")

main()
```

## Question 5 (20 marks)

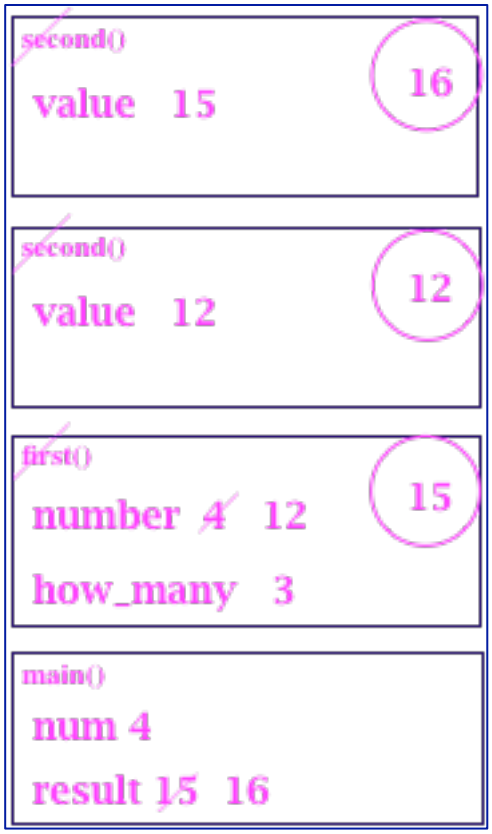
Using the code trace technique taught in lectures, perform a code trace on the following program and show the output.

```
def main():
    num = 4
    result = first(num)
    print("3.",  result)
    result = second(result)
    print("4.",  result)

def first(number):
    how_many = 3
    print("1.",  number)
    number = second(number * how_many)
    return number + how_many

def second(value):
    print("2.", value)
    return value + value % 2

main()
```

The output

```
1. 4
2. 12
3. 15
2. 15
4. 16
```

(6 marks)




(14 marks)