

# THE UNIVERSITY OF AUCKLAND

---

Summer Semester, 2019  
Campus: City

---

## TEST

### COMPUTER SCIENCE

#### 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.

<b>Surname</b>	
<b>Forenames</b>	
<b>Preferred Name (if different to forenames)</b>	
<b>Student ID</b>	
<b>Username</b>	
<b>Monday/Tuesday Lab Times</b>	

<b>Q1</b>	<b>Q4</b>
(/20)	(/20)
<b>Q2</b>	<b>Q5</b>
(/24)	(/20)
<b>Q3</b>	<b>TOTAL</b>
(/16)	(/100)

**Question 1 (20 marks)**

- a) Complete the output produced by the following code.

```
result = 3 * 2 / 6 % 4 - 5 // 2 * 3 % 2
print("Result:", result)
```

```
Result: 1.0
```

(2 marks)

- b) Complete the output produced by the following code.

```
print("Result1:", 17 % 7, "Result2:", 6 % 8, "Result3:", 9 % 3)
```

```
Result1: 3    Result2: 6    Result3: 0
```

(2 marks)

- c) Complete the code below which assigns an
- even**
- number between 50 and 100 (both inclusive) to the variable,
- `even_random`
- . You can assume that the
- `random`
- module has been imported.

```
even_random = random.randrange(50, 101, 2)
```

(2 marks)

- d) Complete the output produced by the following code.

```
phrase = "I am serious; it was a joke"
index1 = phrase.find("a")
index2 = phrase.rfind("suo")
print("1.", index1, "2.", index2, "3.", phrase[6], "4.", phrase[-2])
```

```
1. 2    2. -1    3. e    4. k
```

(2 marks)

- e) Complete the output produced by the following code.

```
phrase = "Rain is saved up in cloud banks"
print("1.", phrase[: 3], "2.", phrase[-3:], "3.", phrase[5: 14: 3])
```

1. **Rai**      2. **nks**      3. **ise**

(3 marks)

f) Complete the output produced by the following code.

```
result = False or True and not True
print("Result:", result)
```

Result: **False**

(2 marks)

g) In the code below, initialise the two variables, value1 and word, so that the output of the code is: Yes.

value1 =

**8**

word =

**"cat"**

(4 marks)

```
if "a" in word and value1 % 2 == 0 and value1 > 7:
    print("Yes")
else:
    print("No")
```

h) Given the following code, what is the type of each of the three Python objects: object1, object2 and object3?

```
a_string = "567821"
number = 56.8
a_list = [7, "4.5", True]

object1 = a_list[0] + number / 3
object2 = a_string[2] * 4
object3 = a_list[2] == False
```

type of object1: **float**  
 type of object2: **string**  
 type of object3: **boolean**

(3 marks)

**Question 2 (24 marks)**

- a) Complete the following program. The program first prompts the user for their username (3 or 4 letters followed by exactly 3 digits) using the prompt "Enter username:". Two example user names are `afer023`, `sli009`.

The program prints the first initial of the username in upper case, followed by ". " followed by the second letter of the username in upper case, followed by ". - " and finally the integer number from the user name ignoring any initial zeroes (the number may be one digit, two digits or three digits). For example, the following screenshots show three **different** executions of the completed program:

Enter username: <b>afer023</b> A.F. - 23	Enter username: <b>sli009</b> S.L. - 9
Enter username: <b>bset976</b> B.S. - 976	

```

user_name = input(prompt)
first_initial = user_name[0]
second_initial = user_name[1]
number = int(user_name[-3:])
to_print = (first_initial.upper() + "." +
            second_initial.upper() + "." + " - " +
            str(number))

print(to_print)

```

(8 marks)

- b) Give the output produced by the following code.

```

number = 23
while number < 39:
    print(number, end = " ")
    number = number + 4

```

```

23 27 31 35

```

(4 marks)

c) Complete the `for ... in range()` loop so that the output of the following code is:

21 26 31 36 41 46 51 56 61 66 71.

```
for number in range(21, 72, 5):  
    print(number, end = " ")
```

(4 marks)

d) Give the output produced by the following code.

```
number = 5  
while number < 15:  
    number = number + 3  
    if number % 2 == 0:  
        print(number, end = " ")  
        number = number + 2  
    print(number, end = " ")
```

```
8 10 13 16 18
```

(4 marks)

e) Complete the output produced by the following code.

```
count_a = 0  
count_b = 0  
  
for number in range(16, 1, -3):  
    count_a = count_a + 1  
    if count_a % 2 == 0:  
        count_b = count_b + count_a  
count_b = count_b + 1  
print("count_a:", count_a, "count_b:", count_b)
```

```
count_a: 5    count_b: 7
```

(4 marks)

**Question 3 (16 marks)**

- a) The first line of the `get_random_word()` function defines the variable, `words`, which is a string made up of words where each word is followed by spaces so that each word takes up **exactly** ten spaces, e.g., the second word "sad" starts at index 10 and ends at index 19, the fourth word "fantastic" starts at index 30 and ends at index 39, and so on. Note: after obtaining the random word from the `words` string, you will need to remove blank spaces from end of the random word.

Complete the `get_random_word()` function which returns a random word (with no blank spaces) taken from the variable, `words`. For example, the textbox below shows a possible output produced when the `main()` function is executed.

```
import random
def main():
    print("1.", get_random_word())
    print("2.", get_random_word())
    print("3.", get_random_word())
def get_random_word()
```

1. so-so
2. dreamy
3. happy

<pre>#Only part of the string, words, is shown here words = "happy      sad      so-so      fantastic dreamy      ... "</pre>
---

<pre>index = random.randrange(0, len(words), 10) word = word_strings[index: index + 10] return word.strip()</pre>
---

(8 marks)

- c) The following program works out car rental costs. The program executes as follows:
- firstly the program prompts the user for the distance travelled (in kilometres),
  - then the program prompts the user for the number of rental days, and
  - finally the program displays a summary including the distance travelled, the number of rental days, the discount amount (if any) and the total cost.

All the functions for this program have already been defined. You are required to complete the `main()` function of the program by adding **FOUR** lines of code, with each line **making a call** to one of the defined functions.

On the next page are two example outputs produced using the completed program (the user input in the examples is shown in a larger bold font).

```
Enter distance travelled: 6050
Enter number of days: 7
Distance: 6050 kilometres
Days: 7 days
Discount: $100
Total: $948
```

```
Enter distance travelled: 100
Enter number of days: 2
Distance: 100 kilometres
Days: 2 days
Total: $55
```

```
def main():
```

```
    distance_prompt = "Enter distance travelled"
    days_prompt = "Enter number of days"
    cost_per_k = 0.15
    cost_per_day = 20

    distance = get_number(distance_prompt)
    days = get_number(prompt_days)
    total_cost = get_cost(distance, days, cost_per_k,
                          per_day)
    display_summary(distance, days, total_cost)
```

(8 marks)

```
def get_number(prompt):
    return int(input(prompt + ": "))

def get_cost(distance, days, cost_per_k, cost_per_day):
    discount = get_discount(distance, days)
    cost = distance * cost_per_k + days * cost_per_day - discount
    return round(cost)

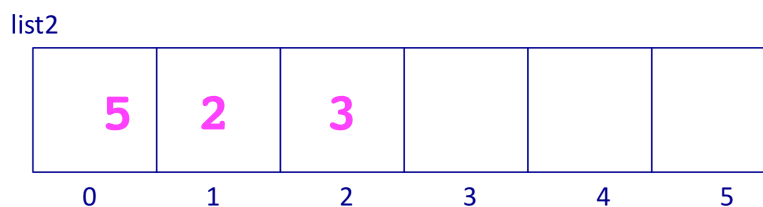
def get_discount(distance, days):
    if days > 10 or distance > 5000:
        return 100
    return 0

def display_summary(num_k, num_days, total_cost):
    discount = get_discount(num_k, num_days)
    print("Distance:", num_k, "kilometres")
    print("Days:", num_days, "days")
    if discount > 0:
        print("Discount: $", discount, sep=" ")
    print("Total: $", total_cost, sep=" ")
```

**Question 4 (20 marks)**

- a) In the boxes below, show the elements of `list2` after the following code has been executed. Use as many of the boxes as you need.

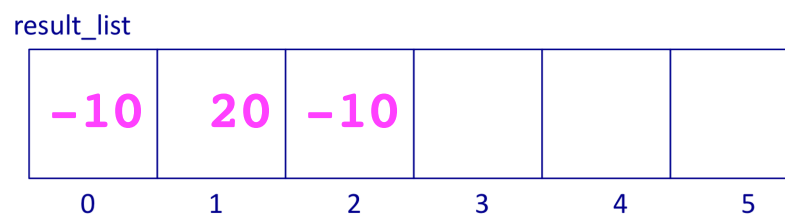
```
list1 = [True, False, False, True, True]
number = 5
list2 = []
for element in list1:
    if element:
        list2 = list2 + [number]
        number = number + 1
    else:
        number = number - 2
```



(6 marks)

- b) In the boxes below, show the elements of the list, `result_list`, after the following code has been executed. Use as many of the boxes as you need.

```
list1 = [1, 2, 3]
list2 = [9, 8, 7]
result_list = [1, 2, 3]
for i in range(0, len(list1)):
    index1 = (i + 1) % 3
    index2 = (i + 2) % 3
    result_list[i] = (list1[index1] * list2[index2] -
                    list1[index2] * list2[index1])
```



(6 marks)



- c) Complete the `display_elements()` function which is passed two lists of numbers as parameters. The function displays a list of all the elements from the first parameter list which are also present in the second parameter list and, on the next line, displays a list of all the elements of the first parameter list which are NOT present in the second parameter list. For example, when the following `main()` function is executed using the completed function the output is:

```
List 1 elements which are in List 2: [2, 3]
List 1 elements which are NOT in List 2: [1, 1]
```

```
def main():
    a_list1 = [1, 1, 2, 3]
    a_list2 = [2, 2, 3, 4]
    display_elements(a_list1, a_list2)
```

```
def display_elements(list1, list2):
```

```
    elements_in = []
    elements_not_in = []

    for element in list1:
        if element in list2:
            elements_in = elements_in + [element]
        else:
            elements_not_in = elements_not_in + [element]

    message1 = "List 1 elements which are in List 2:"
    message2 = "List 1 elements which are NOT in List 2:"

    print(message1, elements_in_list2)
    print(message2, elements_not_in_list2)
```

(8 marks)

**Question 5 (20 marks)**

a) Give the output produced by the following program.

```
def main():
    print("A", end = "")
    do1()

def do1():
    do2()
    print("B", end = "")

def do2():
    print("C", end = "")
    do3()

def do3():
    print("D", end = "")

main()
```

**ACDB**

(6 marks)

- b) Using the code trace technique taught in lectures, perform a code trace on the program on the next page and show the output.

```
def main():
    result = first("DRABBLE")
    print(result)

def first(word):
    part1 = second(word[:2])
    part2 = second(word[2:])
    return part1 + part2

def second(word):
    letters = "ABCDE"
    digits = "35739"

    print(word)

    number = 1
    if word[0] in letters:
        index = letters.find(word[0])
        number = number * int(digits[index])

    if word[1] in letters:
        index = letters.find(word[1])
        number = number * int(digits[index])

    return number

main()
```

Show the output:

```
DR
ABBLE
18
```

```
second()
word "ABBLE" 15
letters "ABCDE"
digits "3579"
number 1 3 15
index 0 1
```

```
second()
word "DR" 3
letters "ABCDE"
digits "3579"
number 1 3
index 3
```

```
first()
word "DRABBLE" 18
part1 3
part2 15
```

```
main()
result 18
```

(14 marks)

**OVERFLOW PAGE**

(If you have used this page, please indicate clearly under the relevant question that you have overflowed to this page)