John Fraser Secondary School
Computer Studies Examination – January 2020

Version Q – Practice Exam (Semester 1)

Name:______________________ **Student #:**___________________

Course: Introduction to Computer Science **Date of Exam:**

Course Code: ICS3U0 **Length of Exam:** 1.5 hours

Teacher: Mr. Seidel/Kowalczewski **Number of pages:** 6 + cover page

Instructions & Notes

1. Read through each question carefully.

2. Write **full sentences** unless otherwise specified.

3. This is an **open book** exam. This means you have access to the following:
    a. Any hand-written notes you brought in with you
    b. Any printed material you brought in with you

4. You do **not** have access to the following:
    a. Your peers
    b. Any electronic device, or internet access

5. List any assumptions as needed for questions you are unsure about, and use the back of the sheets if you need extra space.

6. Each question will only be assessed on the criteria given.

SUCCESS CRITERIA	LEVEL
Can effectively create simple computer programs. (Questions 1, 2, and 3)	
Can effectively test, analyze and document computer programs to ensure accuracy and clarity. (Question 4 and 5)	
Can effectively use the software development life cycle to plan and design computer programs. (Questions 6 and 7)	
Can solve problems related to distinct components of a computer. (Questions 8 and 9)	
Overall Level	

This is a practice exam.

If you attempt this exam, time how long each question takes you to prepare for the exam.

CREATE SIMPLE PROGRAMS

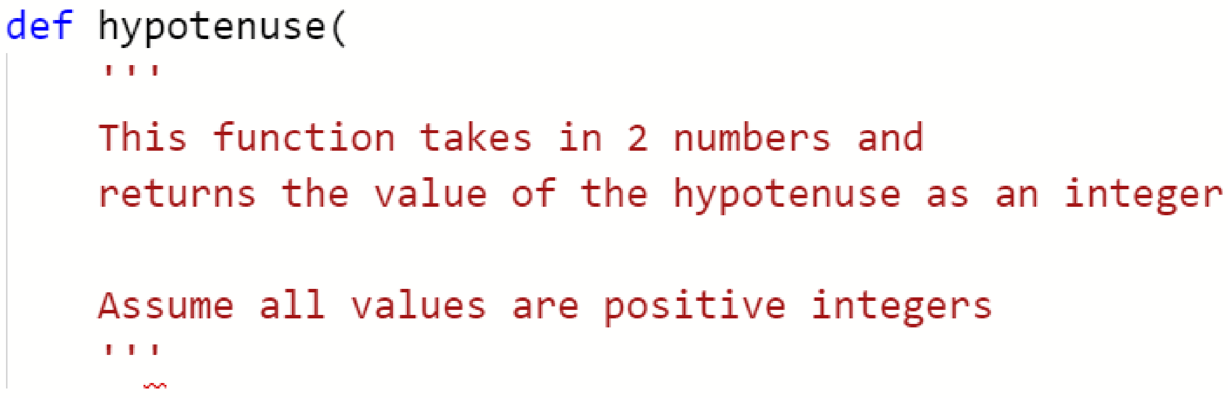
1. Create a small program that:
    a. Reads in information to a list, then
    b. Prints out each item of the list, and
    c. Uses for loops

    Do <u>not</u> include documentation, assertions, logging, or error checking.

    [This question is assessing your ability to use repetition control structures in a program]

2. Finish creating the function below given the limited information provided. **Do <u>not</u> include documentation, assertions, logging, or error checking**.

    [This question is assessing your ability to write custom subprograms, use variables, constants, and selection control structures]



```
def hypotenuse(
    '''
    This function takes in 2 numbers and
    returns the value of the hypotenuse as an integer

    Assume all values are positive integers
    '''
```

This is a practice exam.

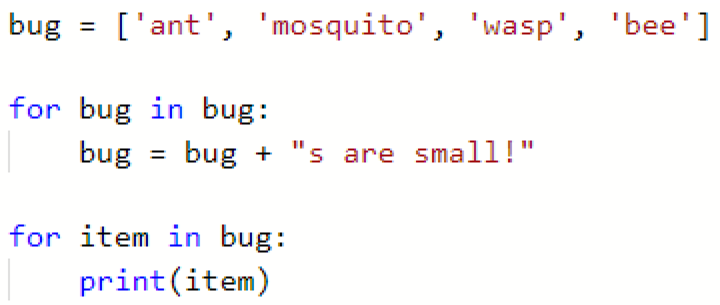
Expected Output

```
ants are small!
mosquitos are small!
wasps are small!
bees are small!
```

3. Given the expected output on the right, which of the following matches. **Choose <u>all</u> that apply.**

[This question is assessing using lists in a program]

A. --

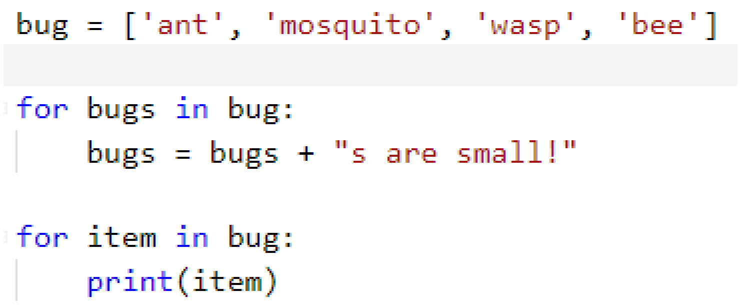


```
bug = ['ant', 'mosquito', 'wasp', 'bee']

for bug in bug:
    bug = bug + "s are small!"

for item in bug:
    print(item)
```

B. --

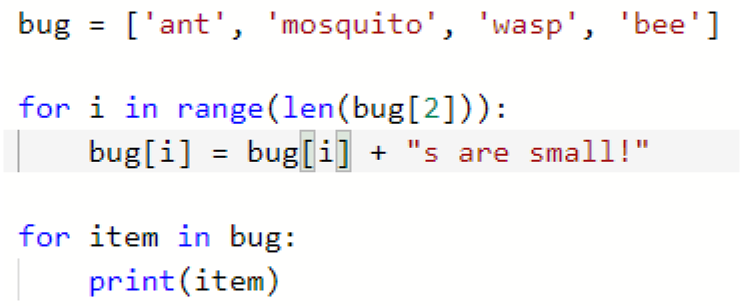


```
bug = ['ant', 'mosquito', 'wasp', 'bee']

for bugs in bug:
    bugs = bugs + "s are small!"

for item in bug:
    print(item)
```

C. --

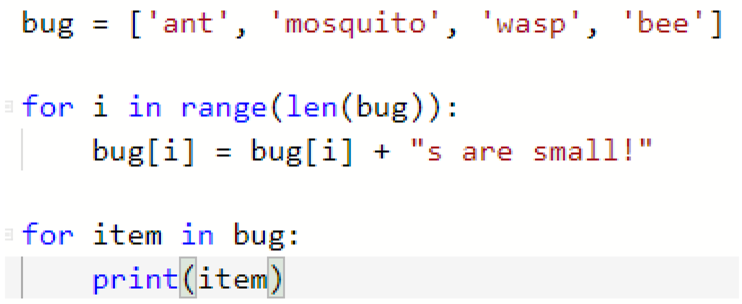


```
bug = ['ant', 'mosquito', 'wasp', 'bee']

for i in range(len(bug[2])):
    bug[i] = bug[i] + "s are small!"

for item in bug:
    print(item)
```

D. --



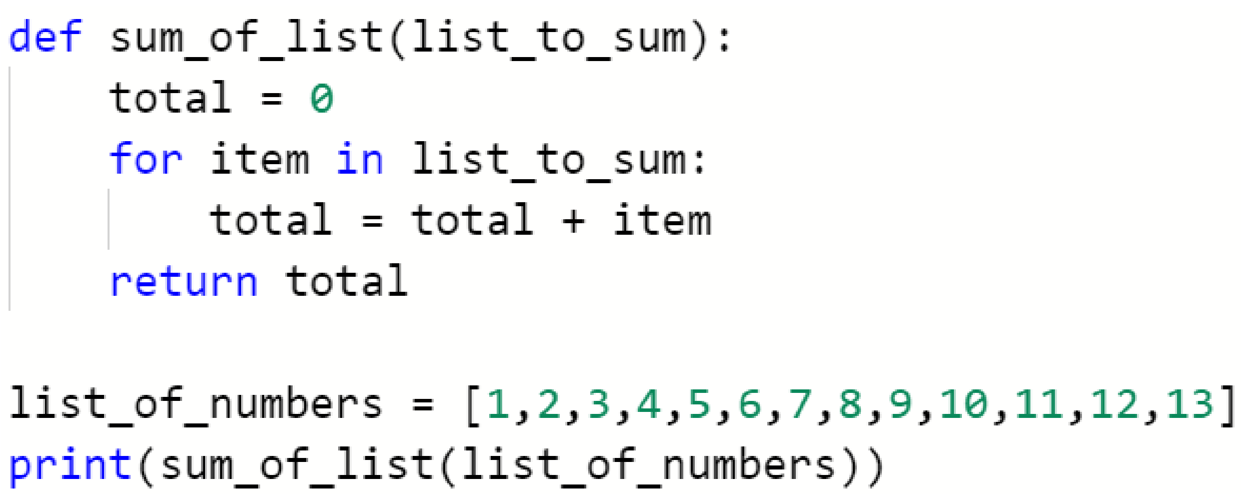
```
bug = ['ant', 'mosquito', 'wasp', 'bee']

for i in range(len(bug)):
    bug[i] = bug[i] + "s are small!"

for item in bug:
    print(item)
```

Test, Analyze and Document

4. Rewrite the function below to include the following:
    a. Documentation, and
    b. Logging
    c. **No exceptions or assertions at all**
    [This question is assessing documenting programs]



```python
def sum_of_list(list_to_sum):
    total = 0
    for item in list_to_sum:
        total = total + item
    return total

list_of_numbers = [1,2,3,4,5,6,7,8,9,10,11,12,13]
print(sum_of_list(list_of_numbers))
```

5. What is the purpose of using exceptions and assertions? Why would industry professionals use exceptions and assertions when they can just use more if statements and print statements?
   [This question is assessing testing and analyzing programs]

SDLC AND DESIGN

6. What other tasks can be done during the design process other than flowcharting and screenshots/designs?
   [This question is assessing your understanding of the software development life cycle]

7. Given the following situation, create a high-level flowchart (10-15 symbols) to represent the situation as you understand it.
   [This question is assessing your ability to create a flowchart]

Program Request

An algae-for-energy company that stewards a multitude of algae wants to regulate the temperature of multiple pools of water they use to store algae and study them to ensure they have a long lifespan and energy production. The company is hiring a software development company (you) to plan out their software requirements. Below are the requirements they have regarding the system:

- Create a phone-based app that can do the following:
  - Monitor and control the temperature of each cooler individually.
  - Monitor the fans and HVAC system (cooling system) for malfunctions.
  - Be able to add algae and update algae information in the database.

PROBLEM SOLVING

8. Describe the differences between "Divide and Conquer" and "Stepwise Refinement". What situations would call for using one over the other?
   [This question is assessing problem-solving algorithms]

9. These questions are assessing organization and backup planning
   a. Why is it important to review your backup plans occasionally?

   b. What happens when your organizational style changes over time?

This is a practice exam.