MEMORIAL
UNIVERSITY

Faculty of Engineering and Applied Science

Department of Electrical and Computer Engineering
St. John's, NL   Canada   A1B 3X5
Tel: 709 864 8177   Fax: 709 864 4042
https://www.mun.ca/engineering/ece

ECE 1020: Introduction to Programming
Mid-term test
15 Mar 2022

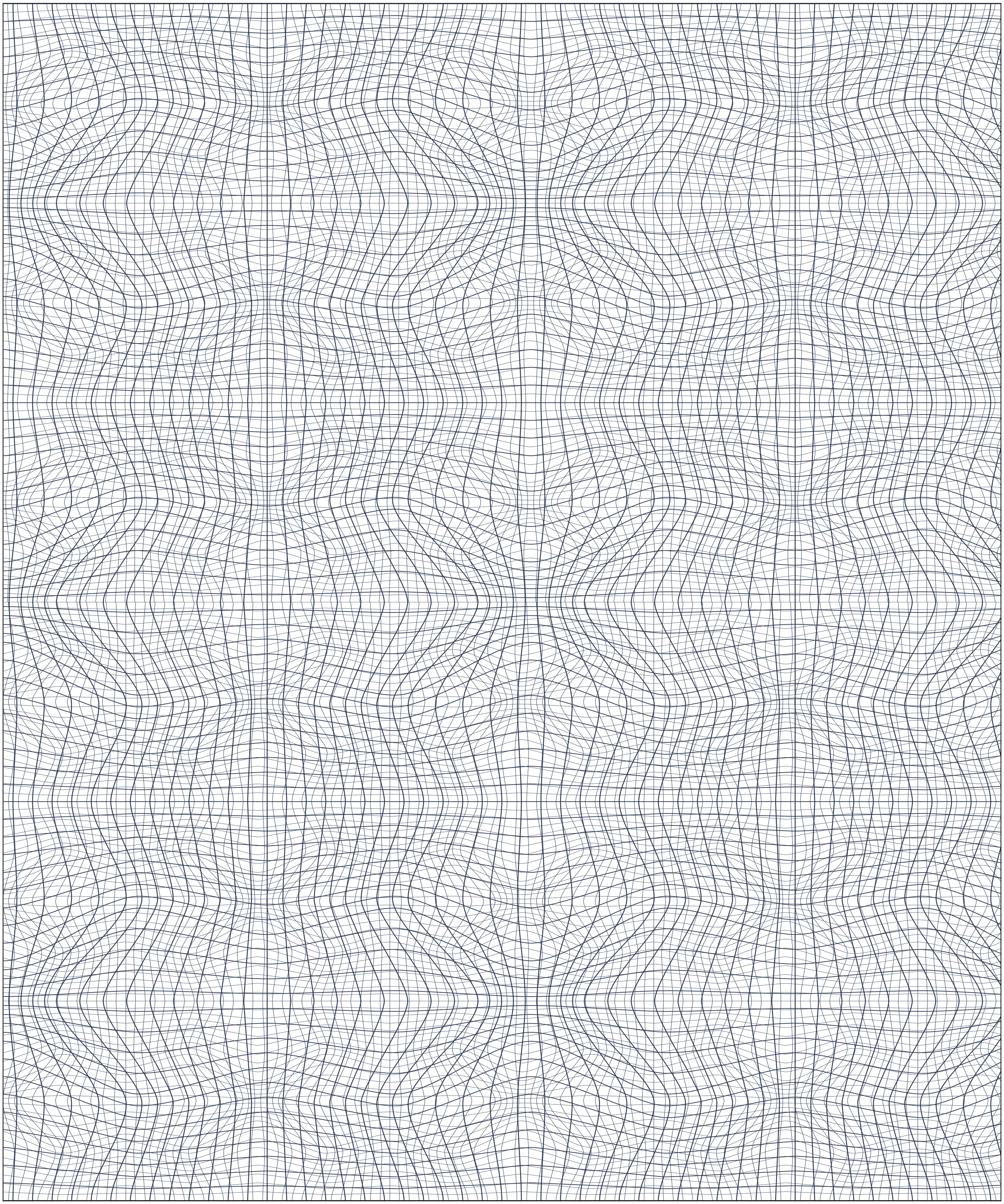
Name:

Student ID:

Instructions

1. Answer all questions.
2. Write your answers in the space provided on this paper.
3. Write your student number at the top of every answer page.
4. This is a closed-book exam: written aids are not permitted.
5. Calculators, phones and all other electronic aids are not permitted.
6. Unless otherwise specified, all code listed in this exam compiles and executes correctly.

Question:	1	2	3	4	5	Total
Points:	10	10	10	8	12	50
Score:						



Programming concepts

1. Identify the programming term that best corresponds to each description. **[10]**

(a) How to do something $\boxed{1}$

✓ **algorithm** ○ argument ○ condition ○ foo ○ scope ○ syntax ○ type

(b) For people $\boxed{1}$

○ algorithmic ○ call ✓ **comment** ○ index ○ parameter ○ syntax ○ type

(c) What to do next $\boxed{1}$

○ argument ○ binary ○ condition ✓ **control flow** ○ expression ○ syntax

(d) A way of representing a procedure $\boxed{1}$

○ compound ○ floating-point ✓ **function** ○ parameter ○ scope ○ type

(e) Accessible everywhere in a file $\boxed{1}$

○ argument ○ assignment ○ call ✓ **global variable** ○ local variable ○ parameter

(f) Name for something $\boxed{1}$

○ expression ○ function ✓ **identifier** ○ index ○ semantics ○ syntax

(g) Like an identifier, but you can't use it as a function name $\boxed{1}$

○ call ○ floating-point ○ integer ✓ **keyword** ○ string ○ type

(h) Mathematics and programming are examples $\boxed{1}$

○ arguments ○ calls ○ methods ○ modules ✓ **synthetic languges** ○ variables

(i) A place to define functions and variables $\boxed{1}$

○ condition ○ expression ○ inclusive `or` ✓ **module** ○ parameter ○ type

(j) Code and data, together $\boxed{1}$

○ Boolean ○ inclusive `or` ○ integer ✓ **object** ○ parameter ○ unary operator

Program analysis

2. Given the following Python code saved in `foo.py`: **[10]**

```
x = 1
y = 2

def foo(f):
    for i in range(3):
        bar(x + f + i)

def bar(n):
    global x

    x += (n + 10) % 5
    y = x + 1

def baz(n):
    print(n)

    if n <= 0:
        return 1
    else:
        return n + baz(n-1)
```

(a) What will the following Python script print when it is run? 4

```
import foo

foo.foo(1)
print(”x:”, foo.x)
print(”y:”, foo.y)
```

Solution: 21

(b) What will the following Python script print when it is run? 6

```
import foo

print(foo.baz(4))
```

Solution:

```
4
3
2
1
0
11
```

3. Errors **[10]**

(a) Are the following identifiers syntactically valid or invalid?

i. `foo` ○ valid ○ invalid 1

ii. `Foo` ○ valid ○ invalid 1

iii. `f00` ○ valid ○ invalid 1

iv. `4oo` ○ valid ○ invalid 1

v. `_foo` ○ valid ○ invalid 1

(b) What is wrong with each of the following?

i. Function call: `foo(x=1, 2)` 1

Solution: Positional argument after keyword argument

ii. First line of a function definition: `def foo(x=1, y):` 1

Solution: Default argument before parameter without default argument

iii. Function definition: 1

```
def foo(n):
    return n - foo(n-1)
```

Solution: Missing base case of recursion

iv. Loop: 1

```
n = 3
while n > 0:
    print(n-1)
```

Solution: Infinite loop

v. Method call: 1

```
s = 'hello!'
indexOf('e')
```

Solution: Missing method target (it's a method, not a function)

Program synthesis

4. Write a Python function that will calculate the sum of the numbers in a list that are divisible by the parameter n. This parameter should default to 2. For example, given the list [1, 2, 3, 4] with no other argument, your function should return the number 6. 8

Solution:

```
def even_sum(l):
    return sum([n for n in l if n % 2 == 0])
```

5. Suppose you were required to write tests for the following Python function: **[12]**

```
def find(s, ch, start=0):
    ”””Find an instance of the character in ‘ch‘ within the string ‘s‘,
    starting looking from the character at the index given by ‘start‘.
    ”””

    # the actual implementation goes here
```

Write six function calls that you would use to test this function and each call’s expected result.

(a) ______________________________ ⇒ ____________________ 2

(b) ______________________________ ⇒ ____________________ 2

(c) ______________________________ ⇒ ____________________ 2

(d) ______________________________ ⇒ ____________________ 2

(e) ______________________________ ⇒ ____________________ 2

(f) ______________________________ ⇒ ____________________ 2