# CSc 110: Practice Exam

## Do not open until the instructor says so!

You should take this exam individually, and you have roughly 50 minutes to complete it. Additional time may be given at the instructor's discretion. Do not cheat off of those nearby you. Make sure to write and/or draw clearly and legibly in all your solutions. Also make sure to structure and style your hand-written code well. Each question has designated location(s) where your answer should go. You may do extra work to arrive at the answer, but the answer *must* go in the designated location. You are not allowed to use a calculator.

| NAME (FIRST AND LAST): |
| --- |

| NETID: |
| --- |

## Question 1

**(A)** Convert the decimal number *117* to binary.

RESPONSE:

**(B)** Convert the binary number *10110* to decimal.

RESPONSE:

**(C)** What does the three-letter acronym *RAM* stand for?

RESPONSE:

**(D)** What does the three-letter acronym *HDD* stand for?

RESPONSE:

**(D)** What is the name of the function that can be used to get a text value from a user?

RESPONSE:

## Question 2 - Evaluate Expressions

Below are multiple short snippets of python code. Each snippet contains a variable named `result`. In the boxes, write both the value and the type of each variable.

### (A)

```
a = 3 * (2 + 3) - (3 * 2)
result = a * 2
```

| VALUE: | TYPE: |
| --- | --- |

### (B)

```
a = True
b = False
c = a and b or a
result = c or b
```

| VALUE: | TYPE: |
| --- | --- |

### (C)

```
a = 7 / 2 + 20
if a > 30:
    a = a * 2
result = a + 2
```

| VALUE: | TYPE: |
| --- | --- |

## Question 3

Below is a python program. You must determine what this program will print out when executed. Put your answer in the area with the "RESPONSE:" label in it. You might find it helpful to use the empty space outside of the answer field to keep track of the values of the variables as the program runs.

```
a = 7
b = 20.0
c = 30.0
while a > 0:
    if b < c:
        print(b, 'is less than', c)
    else:
        print(a)
    c -= 2
    b += 1
    a -= 1
```

RESPONSE:

## Question 4

Below is a python program, with accompanying line numbers. Draw the control flow graph for this code. You can use the line numbers to indicate which lines belong in which boxes. Draw the diagram as clearly as you can in the response box. Also, make sure to include True and False labels where appropriate.

```
 1 print('Welcome to the program')
 2 i = 0
 3 while i < 10:
 4     i += 1
 5     value_a = int(input('Enter value a: '))
 6     value_b = int(input('Enter value b: '))
 7     result = value_a + value_b
 8     print(result)
 9     if result > 100:
10         print(result, 'is greater than 100')
11         print('it is also greater than 50')
12 if i <= 15:
13     print('good')
14 print('bye')
```

RESPONSE:

## Question 5

Take a look at the python program below:

```
1 def calculate_dph(dollars, hours):
2     dph = dollars / hours
3     print('You earned $', round(dph, 2), 'per hour')
4
5 def calculate_ptp(dollars):
6     income = dollars * 50
7     if income > 10000:
8         income = income - (income * 0.1)
9     elif income > 100000:
10        income = income - (income * 0.2)
11    print('Post-tax pay: $', round(income, 2))
12
13 def main():
14    weekly_income = int(input('Enter weekly income: '))
15    weekly_hours = int(input('Enter weekly hours that you work: '))
16    calculate_dph(weekly_income, weekly_hours)
17    calculate_ptp(weekly_income)
18
19 main()
```

Determine what the program will print out for the pair of input values shown below: You should include the lines of code that accept the input. If an error would be produced, write down which line the error would happen at.

### For inputs 200 and 5

RESPONSE:

## Question 6

In this problem you should write a complete python program, not just a single function. Your program should accept a single int input from the user, which you may assume will be an integer between 0 and 100, inclusive. Your program should print out a side-ways pyramid, based on the provided size. The widest point (or highest, if you were to flip it 90 degrees to the left) will be based on the size. You may use both string multiplication and while loops to accomplish this. You should not need to use an if-statement. Below are several examples.

```
Sideways pyramid size: 3
#
##
###
###
##
#
```

```
Sideways pyramid size: 4
#
##
###
####
####
###
##
#
```

```
Sideways pyramid size: 5
#
##
###
####
#####
#####
####
###
##
#
```

RESPONSE:

## Question 7

Below is a table showing made-up/modified weight classes for both men and women for taekwondo sparring. Note: these are not the *actual* and/or *official* weight classes.

| class | male | female |
|---|---|---|
| fly | no greater than 55 kg | no greater than 50 kg |
| middle | greater than 55kg, no greater than 85 kg | greater than 50k, no greater than 70kg |
| heavy | greater than 85kg | greater than 70kg |

Using this table as a reference, You should write a function named `get_class`. This function will take two parameters. The first one will be an integer, which will represent a weight in kilograms (kg). The second will be a string which should be either `male` or `female`. If the input string is not male or female, the function can print `?`. Based on this information, the function should print out the correct class. For example:

- `get_class(50, 'male')` should print `fly`
- `get_class(50, 'female')` should print `fly`
- `get_class(70, 'male')` should print `middle`
- `get_class(70, 'female')` should print `heavy`
- `get_class(80, 'hi there')` should print `?`

Make sure your final answer goes in the response box, on the next page.

RESPONSE: