

Name _____

CSC 221: Introduction to Programming

Sample Test 1

1. True or False?	(6 pts) :	
2. Short Answer	(6 pts) :	
3. Scratch Programming	(10 pts) :	
4. Tracing Code	(12 pts) :	
5. Python Functions	(6 pts) :	
6. Program Design	(12 pts) :	
TOTAL	(52 pts) :	

I pledge that I have neither given nor received unauthorized aid on this test, and that I will not discuss this exam with anyone for 24 hours.

signed _____

1. True or False? (6 points)

_____ If the Boolean expression $(x \neq y \text{ and } y \neq z)$ evaluates to `True`, then the three variables (x , y , and z) must all store different values.

_____ In programming, a *variable* is a name that refers to some value or property (such as the number of heads flipped or the number of seconds in a countdown).

_____ The three major components of the *von Neumann architecture* are the hard drive, RAM, and cache.

_____ In Python, every user-defined function must contain at least one *return statement*.

_____ The Python programming language was invented at MIT as a tool for teaching children the fundamentals of programming.

_____ A *compiler* is a program that takes another program written in a high-level language and translates it into the machine language for a particular computer.

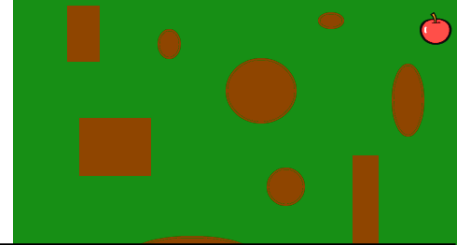
2. Short Answer (6 points)

A. In the first two weeks of this course, you learned the fundamentals of programming by creating Scratch animations. Then, you transitioned into writing Python programs. Describe **two (2)** *significant* similarities between programming in Scratch and programming in Python. Likewise, describe **two (2)** significant differences.

B. Describe **one (1)** *significant* similarity between machines languages and assembly languages. Describe **one (1)** *significant* difference between machines languages and assembly languages.

3. Scratch Programming (10 points)

Consider a Scratch project for moving a bug across the stage. The user maneuvers the bug around, trying to avoid touching the obstacles (i.e., brown shapes), until it reaches the apple. A screen shot of the stage and the scripts for controlling the bug sprite are shown here:



A. What happens *on the stage* whenever the user clicks on the green flag?



B. How does the user maneuver the bug around the screen? For example, if the user wanted to move the bug up the left edge of the screen, what sequence of keystrokes would accomplish this?

C. What happens *on the stage* if the bug ever touches an obstacle (i.e., the color brown)?

D. What happens *on the stage* if the bug ever touches the apple?

E. Suppose we wanted to keep track of the number of times the bug touches an obstacle as it moves around the stage. To allow for this, we have created a variable named # of obstacles. Draw arrows to insert the two blocks below into the appropriate places in the scripts so that the variable correctly keeps track of wall touches.



B. Consider a Python function containing a while loop:

```
def while_trace(x, y):  
    print("START")  
    while x <= y:  
        x += 2  
        y -= 1  
        print(x, y)  
    print("END")
```

- What lines of output would be produced by the call: `while_trace(10, 1)`

- What lines of output would be produced by the call: `while_trace(1, 10)`

- Suppose the last line in the function were indented to the right so that it lined up with the statement above it. Would this change the general behavior of this function? If so, explain how.

5. Python Functions (6 points)

A. Consider a course in which a student's grade is determined by two test scores: a midterm and a final exam. Normally, a student's grade is the average of these two test scores. However, if the exam score is significantly higher (i.e., more than 20 points better than the midterm), then the grade is simply the final exam score. For example, a midterm score of 80 and final exam score of 90 would result in a grade of 85. However, a midterm score of 70 and final exam score of 92 would result in a grade of 92. Complete the definition of the grade function below that computes the student's grade.

```
def grade(midterm, exam):  
    """Calculates and returns the grade using the described method."""
```

B. Consider the following function, which draws a shape using turtle graphics (as in HW2). What shape is drawn by a call to this function? Recall that in Python turtle graphics, the turtle initially faces right.

```
def mystery_shape():  
    t = turtle.Turtle()  
    t.forward(50)  
    t.right(180)  
    t.forward(25)  
    t.right(90)  
    t.forward(25)  
    t.left(180)  
    t.forward(50)
```

6. Program Design (12 points)

Consider a guessing game where the user is asked to guess a number between 1 and 4. The following function could be used to carry out the game:

```
def guess():
    """Allows the user to guess a number between 1 and 4.
       Returns "CORRECT" if they guess correctly, otherwise "WRONG"."""
    pick = random.randint(1, 4)
    user = int(input("Pick a number between 1 and 4: "))
    if (pick == user):
        return "CORRECT"
    else:
        return "WRONG"
```

A. What is the purpose of the call to `int` in the 2nd assignment? Suppose this call were omitted, i.e.

```
user = input("Pick a number between 1 and 4: ")
```

Would this cause an error when the function is called? Would it affect the value returned by the function? Explain.

B. Rewrite the `guess` function below so that it has a parameter, specifying the high number in the range to be guessed. For example, the call `guess(10)` should result in a number between 1 and 10 being picked. Note that the input prompt should include this number as well.

C. Complete the definition of the `many_guesses` function that performs repeated guesses and keeps track of how well the user did. The function has two parameters, the maximum number to be guessed and the number of guesses, and it calls your modified `guess` function that many times. The function should return the percentage (rounded to the nearest integer) that the user guessed correctly. For example, `many_guesses(3, 6)` would result in 6 guesses being made by the user, all in the range 1 to 3. If the user guessed correctly 2 out of the 6 times, then the function would return 33.

Note: specifying 0 for the number of guesses should not cause an error. Instead, the function should return 0 if no guesses were made.

```
def many_guesses(n, num_guesses):  
    """Repeatedly (num_guesses times) picks a random number in the  
    range 1 to n and asks the user to guess it. It returns the  
    percentage the user was correct (rounded to the nearest int)."""
```