CSC 2280
Introduction to Computer Science
Florida Southern College

Assignment 1
Due: Monday, February 4, 2019

1. True/False (10 points)

False Computer science is the study of computers.

False Programs no longer require modification after they are written and debugged.

True Python identifiers must start with a letter or underscore.

False Secondary memory is also called RAM.

False Type conversion functions such as `eval` are completely safe for getting numeric user input whereas `float` and `int` can be unsafe.

2. Multiple Choice (10 points)

A In mixed-type expressions involving `floats` and `ints`, Python will convert:
(a) `ints` to `floats` (b) `ints` to `strings`
(c) `floats` to `ints` (d) both `ints` and `floats` to `strings`

B Which of the following is *not* a part of the IPO pattern?
(a) output (b) program
(c) input (d) process

C Which of the following is *not* a built-in operation?
(a) `str()` (b) `help()`
(c) `floor()` (d) `%`

B Which of the following is a legal identifier?
(a) `123abc` (b) `hAm_And_chEEsE`
(c) `my variable` (d) `Cel$iuS`

A Which of the following is *not* true of comments?
(a) They make a program more efficient.
(b) They are intended for human readers.
(c) They are ignored by Python.
(d) In Python, they begin with a pound sign (`#`).

3. Understanding Code (20 points)

Download the Python code for the *Birds of a Feather* solitaire game from Portal. First, have fun playing the game by running the play.py file in your favorite IDE or by going to a command prompt and typing:

```
cd ~/Documents/python/BirdsOfAFeather
python play.py
```

The first line of code changes your current directory to the BirdsOfAFeather directory (NOTE: your directory name may be slightly different depending on where you downloaded the code to). The second line of code runs the play.py file using Python.

After trying to play the game, examine the code and write your answers to the following questions:

(a) Identify 3 separate lines of code that <u>you are familiar with</u> and briefly describe in your own words the purpose of each line. Each of your selections MUST be different (i.e., do not simply use 3 `print` statements!). Please include line numbers.

Line 16: print(node) - This line prints values of the variable node

Line 68: for j in range(i + 1, num_cards): - this line begins a for-loop

Line 9: def play(): - this line is what holds the value of the rest of the code.

(b) Identify 3 separate lines of code that <u>you do not understand</u> and briefly describe why each line is interesting to you or what you think the line of code may be accomplishing. Please include line numbers.

The elif command is used frequently throughout the code and I have no idea what it does, but I think it may be an alternative form of the if command

Line 88 – is_legal_play = False – I think this could be a player command to ask if a play is legal… maybe?

Line 14 – while True: - I think this command could be used to create certain results so long as what follows it remains true.

4. Error Finding (20 points)

Debugging code is an important and useful skill. For this problem, your task is to identify 5 distinct errors in the code below and briefly explain each error. There are actually more than 5 errors, but you only need to identify 5. Note that errors may relate to both syntax and semantics. HINT: trying to run the code can sometimes help identify errors.

```
# convert.py
# This is a program to convert pounds to kilograms.

define main()
    print("Convert pounds to killograms")
    unicorns = input(Enter a number in pounds: )
    how_many_lbs = unicorns * 2.20462
    print(unicorns, "lbs =", how_many_lbs, "kg)

Main()
```

Error #1 – The main at the end should not have a capital M
Error #2 – No close quotation marks on kg in line 5
Error #3 – Kilograms is misspelled in line 2
Error #4 – Needs to be a colon in line one after the parenthesis
Error #5 – Define should be define

5. Programming (40 points)

(a) A Fibonacci sequence is a sequence of numbers where each successive number is the sum of the previous two. The classic Fibonacci sequence begins: 1, 1, 2, 3, 5, 8, 13,.... Write a Python program (called fibonacci.py) that computes the first *n* numbers in the Fibonacci sequence, where *n* is a value input by the user. For example, if *n* = 6, then the result is 1 1 2 3 5 8. Print the result to screen with a single space between the numbers in the sequence.

(b) Write a Python program (called calculate_tip.py) that calculates three different tips on a dinner check amount: 15%; 18%; and 20%. Where does the check amount come from? The program should prompt you to enter it. When you do, the program will read it and store it in a variable. Your program should then calculate the tip amounts and print them to the screen.

A few things to keep in mind:

- You should include comments in your code. Recall that these act as a form of internal documentation.
- Pay attention to the prompts and output lines. Well-designed programs are easy to use, descriptive, and self-explanatory.
- When you read input from the user in Python, it is read in as a string (a sequence of characters) rather than a number. You will have to convert it to a number (this is known as type casting). To type cast a variable called x to a real number, you use this: `float(x)`