Exam 1 Review Worksheet

CS/ECE 252 Section-2 (MWF 11:00)

Distributed on Wednesday, September 23rd
Discussed on Friday, September 25th
Exam on Monday, September 28th

Topic List

Chapter One

- What is a computer?
    - Types of computers
        - Stored program
        - Fixed program
    - Functions of a computer
        - Arithmetic
        - I/O
        - Branching
        - Storage
- Abstraction
    - Benefits of abstraction
    - Layers of abstraction in a computer
        - ISA
        - Programming language
        - State machines
        - Microarchitecture
        - Logic gates
        - Transistors
- Binary representation of numbers
- What is an algorithm?
    - Properties of algorithms
- Types of errors
    - Syntax
    - Logic
    - Runtime
- State machines
- Theory
    - Computational complexity
    - Turing machines/Turing complete

Chapter Two

- Programming
- Efficiency vs scalability
- Python basics

- Syntax
  - Valid variable names
- Conditionals
  - IF/ELSE statements
  - WHILE statements
- Variable assignments
- Mathematical expressions
- Strings
- Print function
- Comments
- Factorial example
  - Off-by-one errors

Chapter Three

- Readability vs. maintainability
- Arrays
  - Creating an array
  - Accessing elements in arrays
  - Strings as arrays
- Logical operators
  - >, <
  - ==
  - !=
  - and
  - or
- Functions
  - Defining a function
  - Returning values from functions
- Scope

Homeworks 1-3

1. Logic gates is a level of abstraction between microarchitecture and transistors. How exactly does logic gates connect microarchitecture to transistors?

2. Recall the layers of abstraction involved in computing, from the very highest **system** level all the way down to the **transistor** level. What layer of abstraction is described in each of the following statements?

2.1 A bunch of loads into memory, stores from memory, arithmetic computational instructions, and a branching instruction at the ISA level, is all contained in one single Python "if" statement.

2.2 This contains both the architecture of a computer as well as the instruction set for each different component.

2.3 An ALU consists of many AND and OR gates.

2.4 A very common type of 2-input NAND gate consists of 4 transistors.

3. What role does the number sign, pound sign, or hashtag character (#) play in Python? Give an example of using it.

4. Step through this code by filling out the table below.

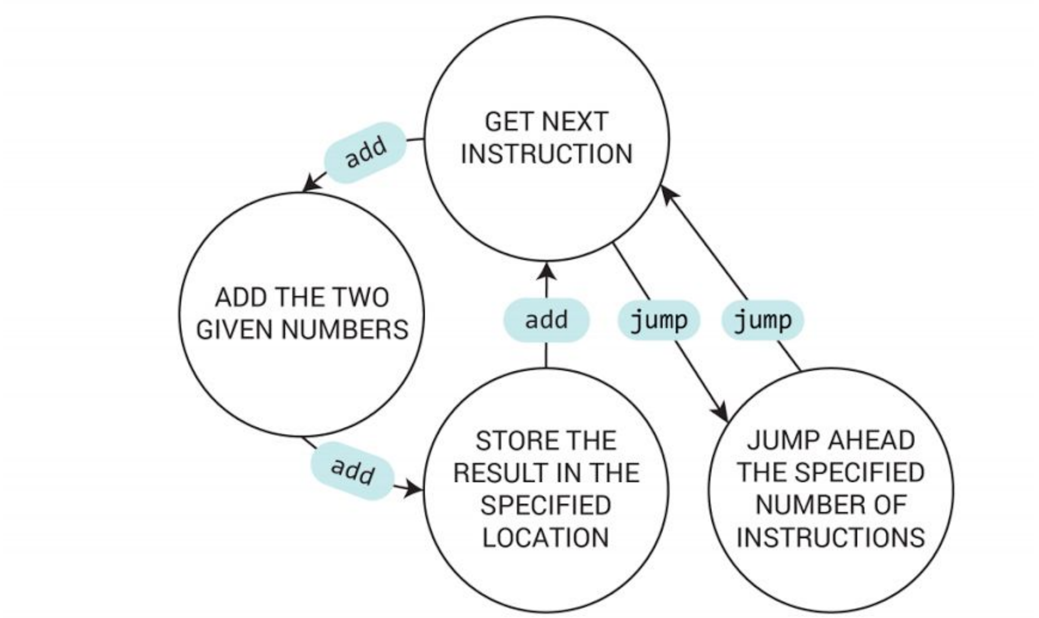
```
(i)   x = 13
(ii)  while(x > 5):
(iii)      print("hello")
(iv)       x = x - 3
```

Current line	Calculation performed by this line	Variables	Next line:
		x	
i	initializing x to 13	3	ii

5. Give a counterexample to this statement: *Using only letters, numbers, and underscores will always form a valid variable name.*

6. You are given three variables *a*, *b*, and *c*. Write a program to sort them ascending, such that *a* will be less than or equal than *b,* which will be less than or equal to *c*. Feel free to use Homework 2 Question 10 (write a program to swap values of two variables) to assist you. You may create additional variables if needed.

7. Let say this is our state machine.


GET NEXT INSTRUCTION
add
ADD THE TWO GIVEN NUMBERS
add
jump
jump
add
STORE THE RESULT IN THE SPECIFIED LOCATION
JUMP AHEAD THE SPECIFIED NUMBER OF INSTRUCTIONS

Also, let's say that our machine is currently in the "GET NEXT INSTRUCTION" state. What state will our state machine be in after add, add, add, jump, jump, add, add?

Action	*start*	*add*	*add*	*add*	*jump*	*jump*	*add*	*add*
State	GET NEXT INSTRUCTION	ADD THE TWO GIVEN NUMBERS						

8. Let’s say a piece of code does not work as expected. How do you determine whether it is a syntax error or a logic error?

9. (a) What is abstraction? Why is it important?

(b) Give a brief description of the following layers of abstraction:

i. programming language

ii. microarchitecture

iii. ISA

10. What is an algorithm? What are trade-offs when it comes to writing an algorithm?

11. What is a computer? What is the difference between a stored-program device and a fixed-program device?

12. Write a program to compute the factorial of 4. Your final printed answer should be 24. You may create as many variables as needed.

13. Rewrite the following piece of code using an array

```
temp1 = 72.7
temp2 = 74.6
temp3 = 68.9
temp4 = 70.2
temp5 = 76.1
print("local temperatures:")
print("Station 1: " + temp1)
print("Station 2: " + temp2)
print("Station 3: " + temp3)
print("Station 4: " + temp4)
print("Station 5: " + temp5)
```

14. Write a function that accepts two numbers and return the product of the two numbers.

15. Create a 5 x 5 multiplication table. Use the function you created in question 14 to do the multiplication. Store the products in a two dimensional array (an array of an array), and print the two dimensional array at the end.

17. Print the outputs of the following piece of code.

```
def foo(x):
   print(x)
   x = 5
   return x
y = 3
print(foo(y))
print(y)
```