

CS 150 Spring 2014
Test #2 - PRACTICE
April 11th, 2014

Name:

This test is closed notes, closed book. No aids are permitted. In total there are 5 questions. You have 50 minutes. You may use the back of any page if you need more space; indicate on the front of that page if you do so.

Good luck!

Question 1	/25 pts
Question 2	/15 pts
Question 3	/15 pts
Question 4	/20 pts
Question 5	/25 pts
Total	/100 pts

Honor Code:

1. (25 pts) **Short Answer.**

(a) Write the binary (base 2) number 1001 in decimal (base 10).

(b) Write the decimal number 53 in binary.

(c) What will the output of this code be?

```
A = [1,2,3]
B = A + [4]
C = A.append(5)
print(A, B, C)
```

Suppose A , B and C are boolean variables. Determine whether each of the following statements are true or not, and justify your answer. (If False, give a counter example)

(d) $A \text{ AND } (B \text{ OR } C) = (A \text{ AND } C) \text{ OR } (A \text{ AND } B)$

(e) $\text{NOT } (A \text{ OR } B) = (\text{NOT } A) \text{ AND } (\text{NOT } B)$

2. (15 pts) **Lists**

Write a Python function called `min(A)` which takes in a list `A`, and returns the smallest element in that list. (Do not use the built in `min` list function.) You can not assume the list elements are any particular type, but you can assume that `<`, `<=`, `==`, `!=`, `=>`, `>` work.

3. (15 pts) **Recursion.** Consider the following recursive function.

```
def A(x) :  
    print(x)  
    if (x == 0) :  
        return 1  
    else :  
        r = x//2  
        return 1 + A(r)
```

Specify the output and the return value for each of the following function calls.

output

return value

(a) A(5)

(b) A(6)

(c) A(13)

4. (15 pts) **More Recursion.** Write a function `sum(A)` that takes a list of integers and RECURSIVELY adds together all the integers, returning a single integer value.

5. (15 pts) **Classes and Objects.** Suppose you want to create a class called `Pirate` to represent pirates. Each instance of `Pirates` should include fields to store the given pirate's name, how many gold `doubloons` he has, and a boolean value indicating whether or not he has a `pegleg`. The `Pirate` class should also keep track of how many pirates have been created.

(a) Write a class definition for `Pirate`, with a constructor that takes in the name of the pirate, and if he has a peg leg. The constructor should also have an argument `doubloons`, with a default value of 0.

(b) Write a function called `rob` that takes in another pirate `p` and takes away the `doubloons` belonging to `p` and gives them to the pirate calling the method.

(c) Write a client program to do the following: Create a pirate named "Pegleg Pete" who has a peg leg and 0 `doubloons`, and a pirate named "Hookhand Harry" who does not have a peg leg, and has 50 `doubloons`. Have Pegleg Pete rob Hookhand Harry.

(d) Define a subclass of `Pirate` called `Captain` that includes the name of the pirate's ship, the ship's maximum speed, and a boolean value indicating if the ship has sunk. The class `Captain` should override its parent's constructor. The new constructor should take in the pirate's name, peg leg status, ship's name, maximum speed of ship, and optional number of doubloons. The constructor should first call the constructor for `Pirate`, but then add new fields to add the ship's name, speed, and sunkenness.

(e) Define a new `Captain` function called `attack` that takes in another `Captain` `seadog` and sinks his ship.