University of Cape Town ~~ Department of Computer Science

Computer Science 1015F ~~ 2015

Class Test 2 draft

***** Solutions *****

Question 1 - Functions [10]

Blah

The solution

Question 2 - Hardware and Software [9]

1. Blah [3]

Solution

2. Blah [2]

Solution

Question 3 - String Problem Solving [6]

Blah

Code

Solution

Question 4 – Arrays and Testing [20]

Examine the `test2_2016.py` module listed on the last sheet of the test and answer the following questions.

a) Write down the **exact output** when this module is executed (e.g. when the user presses the "Run" button in Wing101). [4]

[2, 4] #[1] mark
[4] #[1] mark
['d', 'o', 't'] #[1] mark
[] #[1] mark *[4]*

b) You have been asked to test the function `daFunc()` using either an **exhaustive testing** or a **random testing** strategy. Which method do you choose and why? [2]

random testing (1 mark) . Exhaustive testing would require infinitely many input values (1 mark) – all possible pairs of lists. *[3]*

c) Does the function call

```
daFunc([2,3],[1,4])
```

provide **statement coverage** testing of function `daFunc`? Explain your answer. [2]

Yes. [1] All lines in the programme are translated and executed.[1]

d) Does the function call

```
daFunc([2,3],[1,4])
```

provide **path** testing of function `daFunc`? Explain your answer. [2]

No. [1] An empty list for x will result in the loop not executing [1] [or another valid alternative path]

e) Each of the modules below will generate an error. In each case, explain **both** what **kind** of error it is and how to **fix** it. [5]

A.

```
import Test2_2016
print(Test2_2016.daFunc([2 3 4],[2 5 6]))
```

Syntax error (or compile-time error) [1] – fix by putting commas between the list elements.[1]

B.

```
import Test2_2016
print(Test2_2016.daFunc([2,3],[3]))
```

Runtime error (or list indexing error) [1] – fix by checking that lists x and y are of equal length (or y is not shorter than x) before entering loop [2] #othre valid fix OK too.

f) Rewrite the code for the function `daFunc(x,y)` so that it works as follows. This function should a new list containing all the elements of list `x` that also occur in list `y`. For example, in the Python3 interpreter:

```
>>>daFunc([2,3,5],[5,3])
   [3, 5]
>>>daFunc([2,3,5],[6])
   []
>>>daFunc(['a','b','c'],['c','b','a'])
   ['a', 'b', 'c']
```

Complete the code below: [5]

```
def daFunc(x,y):
```

One correct answer is:

```
def daFunc2(x,y):
    z=[]  #[1] mark
    for i in x:  #[1] mark
        for j in y: #[1] mark
            if i==j: #[1] mark
                z.append(i) #[1] mark
    return z # this must be here - [-1] if not.
```

Code examples for the test – you may detach this sheet.

Question 4

```
#test2_2016.py
def daFunc(x,y):
    z=x
    for i in range(0,len(x)):
        if x[i]>y[i]:
            z[i]=x[i]
        else: z[i]=y[i]
    return z

print(daFunc([2,3],[1,4]))
print(daFunc([4],[1,5,6]))
print(daFunc(['c','a','t'],['d','o','g']))
print(daFunc([],[20,30,40]))
```