University of Cape Town ~~ Department of Computer Science

Computer Science 1015F ~~ 2018

Class Test 2

Marks : 35

Time : 40 minutes

Instructions:

1. Answer all questions.
2. Write your answers in PEN.
3. All questions refer to version 3 of the Python programming language.

Question 1 [8]

Examine the Q1.py module listed below and answer the following questions.

```
#Q1.py
print("--Welcome to the sandwich constructor--")
print("(Other than bread) tell us what you want")
ing = input("Another ingredient? Y/y/N/n")
while ing== 'Y' or 'y' or 'N' or 'n':
    ing = input("What ingredient do you want?")
    ing = "{:$^11}".format(ing)
    print("Okay, we'll add",ing)
    #ing = input("Another ingredient? Y/y/N/n")
```

(a) Does the last line being commented out make the `while` loop an infinite loop? Why or why not?

No [1] is infinite whether it's commented out or not[1]
because strings (e.g.) "y" always evaluate to True [1] thus
the condition or <string> will always be True [1] [4]

(b) What *exactly* would line 6, `ing = "{:$^11}".format(ing)` do if `ing` = "cheese"?

ing = "$$cheese$$$" [2] if perfect, else [1] if have at least 1 '$' before and after 'cheese' [2]

(c) What are the purposes of the `break , continue` statements?

Break exits immediately from a loop [1]
Continue immediately starts the next iteration of the loop [1]

[2]

Question 2 [17]

(a) Write a function, `ingredient_rev_up`, that reverses the (approximately) first half of an ingredient string *character by character using the length of the ingredient string*, and converts the 2nd half into uppercase and then returns the reversed ingredient

```
def ingredient_reverse(ing):
    revup = ""
    for i in range(len(ing)//2,-1,-1):
        revup += ing[i]
     for i in range (len(ing)//2,len(ing))
        revup += ing[i].upper()
    return rev
```

[1] for def ingredient_reverse(ing) [-1 if no argument]
[1] for return
[1] for use of len//2 (note integer division)
[5] for range statements [[1] each for start,stop and step values]
[1] for rev += ing[i]
[1] for rev += ing[i].upper()
[-1] for any indentation errors

[11]

(b) What is the missing line to complete this program so it is not an infinite loop?

```
j=10
while j != 5:
  j-=1
  print (j)
  if j==5:
     # MISSING LINE
  j-=1
```

[2]

Break[2]
Or j=6[1 mark]

(c) What is the output of the following program?

```
def Funky(x,y,z):
    if x>y:
      if y>z:
        print(x y z)
      else:
        result=c
    if x==y or y==z or z==x:
      print("All values are equal!")

Funky(7,5,1)
```

```
Funky(5,5,9)
```

[4]

7 5 1 [2]

```
All values are equal! [2]
```

Question 3 [10]

(a) What is the difference between primary and secondary storage? Give an example of each.

Primary storage: data not lost when PC switched off [1]
e.g. RAM [1]
Secondary storage: data accessible only when PC on [1]
e.g. HDD [1

[4]

(b) Why are programming languages (e.g. Python) more suitable for writing programs than natural (human) languages such as English?

Human languages ambiguous and imprecise [1]
Programming languages precise syntax and semantics[1]

[2]

(c) Name *two* ways to convert a program from a high level language (e.g. Python) to machine language. Give the differences (if any) between these two ways.

Compilation and interpretation [2]
Compilation: a program that translates a program one-shot .
Only reads the source code once. [1]
Interpretation: a program that analyses and executes
instruction by instruction as necessary. Need it every time
the program is to be run [1].

[4]