## University of Cape Town ~~ Department of Computer Science

## Computer Science 1015F ~~ 2014

# Class Test 2

***** Solutions *****

**Enter the following details AND shade in the corresponding blocks to the right with your Student Number.**

**Faculty (please tick one):**

| Science | Engineering | Commerce | Humanities | Other: |
|---|---|---|---|---|

**Student Number :** ______________________________

**Name (optional) :** ______________________________

**Marks : 35**

**Time : 45 minutes**

**Instructions:**

a) Answer all questions.

b) Write your answers in PEN in the spaces provided.

c) Show all calculations where applicable.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| A | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | 0 |
| B | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | 1 |
| C | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | 2 |
| D | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | 3 |
| E | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | 4 |
| F | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | 5 |
| G | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | 6 |
| H | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | 7 |
| I | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | 8 |
| J | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | 9 |
| K | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| L | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| M | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| N | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| O | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| P | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| Q | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| R | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| S | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| T | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| U | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| V | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| W | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| X | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| Y | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |
| Z | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | | | | |

**FOR OFFICIAL USE ONLY:**

| **Question** | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Max** | | 15 | | 10 | | 10 | | | | | | | | | | | |
| **Marks** | 0 | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| | 1 | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| | 2 | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| | 3 | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| | 4 | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| | 5 | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| | 6 | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| | 7 | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| | 8 | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| | 9 | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| **Marker** | | | | | | | | | | | | | | | | | |

**Question1 [15]**

Examine the Q1.py module listed on the last sheet of the test and answer the following questions.

(i) What is Unicode? [1]

*Unicode is a code for representing characters (binary) as numbers. [1] It includes all the ASCII characters plus many more exotic characters..*

(ii) What do the `ord()` and `chr()` functions (used in the Q1.py module) return? [2]

*The ord function returns the numeric (Unicode) value of a character [1]*
*The chr() function return the character corresponding to the Unicode [1]*

(iii) Explain clearly, **in English** and in general terms, what the `fudge` function returns. [2]

*The munge function shifts the individual characters of a word along by 3 through the alphabet (or, really, through Unicode) and returns the altered word. If the shift runs over the end of the alphabet, it starts from the beginning again. [2]*

(iv) Write down the exact output of the `Q1.py` module if the user runs the module. [3]

*ere #[1]*
*cls #[1]*
*Alphabetical characters only! #[1]*

(v) Write the code for the function `wrdTotal(wrd)` so that it works as follows. This function should **return** the number of different characters in `wrd`, or zero if `wrd` is not a string. For example (in the Python3 interpreter):

```
>>>wrdTotal("banana")
>>>3
>>>wrdTotal(14)
>>>0
>>>wrdTotal("aaaaaaaa!")
>>>2
```

[7]

```
#there are many ways to do this - clever tricks OK.
def wrdTotal(wrd):
    no_chrs=0 #[1]
    if type(wrd)==str:     #[1]
        while wrd:  #[1] for loop of sorts, if necessary)
            no_chrs+=1  #[1]
            wrd=wrd.replace(wrd[0],'') #[2] for removing characters somehow, can use a list
etc.
    return no_chrs  #[1] for returning!
```

**Question2 [10]**

Study the following program to find the position of the first occurrence of an item in a list:

```
def index (values, item):
    for i in range (len (values)):
        if values[i] == item:
            return i
    return -1

item = int (input ("Enter an item:\n"))
print (index ([1,2,3,4], item))
```

(i) Suppose that we are using equivalence classes to test the program. Describe the 2 equivalence classes and 2 boundary values that can be used when testing the function. DO NOT provide test values – only descriptions. [4]

*within list [1]; outside list [1]; first item [1]; last item [1]*

(ii) Provide a set of test values that will test this program when using path testing (Hint: you need 2) [1]

*any value within list [1/2]; any value outside list [1/2]*

(iii) How many test values are needed when using exhaustive testing? [1]

*infinitely many [1]… OR as many as there are integer values [1]*

(iv) If one of the quotation marks was missing, what type of error is this: logic or syntax? Does it get detected at compile-time or at run-time? [2]

*syntax [1]; compile-time [1]*

(v) What are 2 techniques that may be use by a programmer to find the cause of logic errors in a program? [2]

*trace statements [1]; using a debugger [1]*

**Question3 [10]**

Index functions that are able to search for an item in a list are one of the most useful programming constructs. With reference to the index function provided in the previous question, answer the following questions.

(i) Write an index function that returns the position of the LAST occurrence of an item in a list. Assume that the list and the item are passed in as parameters. If the item is not found, return -1. (Hint: modify the program in Question 2) [3]

*one possible solution:*

```
def index (values, item):
   for i in range (len (values)-1,-1,-1): [2 marks for range; 1 mark for same rest of function]
       if values[i] == item:
          return i
   return -1
```

(ii) Write an index function that prints out the position of EVERY item in a list that matches a given item. Assume that the list and the item are passed in as parameters. If no matching items are found, print out "No items match". [7]

*one possible solution (assume they may use any algorithm ... and use negative marking: -1 for each error in their program)*

```
def index (values, item):
   found = False
   for i in range (len (values)):
    if values[i] == item:
        found = True
        print (i)
if not found:
   print (“No items match”)
```

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

## Question 1

```
#Q1.py
def fudge(wrd):
     if wrd.isalpha():
          wrd= wrd.lower()
          fdgedWrd ="" #empty string
          for i in wrd:
               x=ord(i)
               x+=3
               if x>ord('z'):
                    x-=26
               fdgedWrd+=chr(x)
          return fdgedWrd
     else:
          return("Alphabetical characters only!")

def wrdTotal(wrd):
     pass
     #function to be rewritten

y= fudge("Bob")
print(y)
y= fudge("zip")
print(y)
y= fudge("have2have")
print(y)
```