University of Cape Town ~~ Department of Computer Science

Computer Science 1015F ~~ 2017

Class Test 2

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 :

MBRKAB001

Name (optional) :

[redacted]

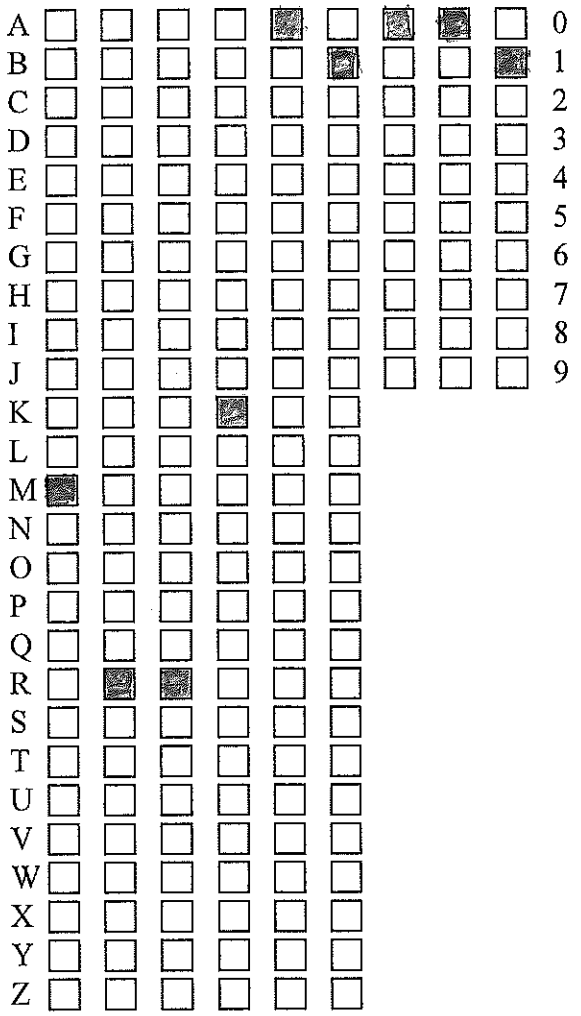
Marks	**: 35**
Time	**: 40 minutes**

Instructions:

a) Answer all questions.

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

c) Show all calculations where applicable.



FOR OFFICIAL USE ONLY:

Question	1	2	3	4	5	6	7	8
Max	10	7	6	12				
Marks	2	6	0	6				
Marker								

38

Question 1 - Functions [10]

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

(i) Implement the missing `prepare_alexa_string` function using string formatting and a docstring. The function should return a string that Alexa will eventually read out that says her favourite country is the `country` argument provided. For example, passing in the string "South Africa" should return the string "My favourite country is South Africa!". Likewise, calling `prepare_alexa_string ("Zimbabwe")` should return the string "My favourite county is Zimbabwe!" [4]

```
def prepare_alexa_str (country):
```

if fav_country is Y

return fav_country

(ii) Assume that the return value of the function `alexa_fns.get_fav_country()` is "South Africa" and you have correctly implemented the function above. What is the output when the code from the `Q1.py` module is executed? [2]

Beryl South Africa

(iii) What is the purpose of the following line? [2] ½

```
if __name__ == "__main__":
```

The purpose of this [illegible] line is [illegible] ½ when the program runs each [illegible] "__main__" program

(iv) Rewrite the `main()` method to be a single line without changing the behaviour. [1] 0

(v) Give one reason why **global variables** are a bad idea. [1] 0

Global variables are saved be modified or changed and so when running a program it will always give the output of the global variable, cc

½

Question 2 - Hardware and Software [7]

(i) In the early 1800s, the Jacquard Loom used punch cards to control a loom in order to simplify the process of manufacturing complex designs for textiles. While this is not a general purpose computer, it is an example of an early computing device. The Jacquard system contains the equivalent of a CPU and Machine Code. Identify which part of the Jacquard loom corresponds to each of these. [2] 2

- The machine code was the punch cards ✓
- The CPU was the loom of the ✓

(ii) Give **two differences** between high-level and low-level languages. [2] 2

High-level languages are easy for humans to understand while low-level languages are easier for the computer to understand. Low-level languages also ✓✓

(iii) Von Neumann's Architecture described memory, a control unit, and what other elements? [1] 1

An Arithmetic logical unit as well as an input and output. ✓

(iv) Give **one advantage** and **one disadvantage** of **assembler**. [2] 1

- Advantage - Very fast to execute the program ✓
- Disadvantage - Very slow to write the program.

6

Question 3 - String Problem Solving [6]

Write a function that takes in a string as an argument and switches the case of each alphabetic character. Your method should replace all of the lowercase letters in the original string with uppercase and replace all uppercase letters in the original string with lowercase. Other characters should remain unchanged.

You may use other string functions, such as `upper()` and `lower()`, character-based processing and/or string slicing as needed.

Your function will take a single argument: the original source string.

```
def convert_case (source):
```

Question 4 – Testing [12]

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

(vi) **From the MK.py module**, give an example of:

A. A syntax error [2] 0

result =c X

B. A runtime error [2] 1✓

else: else: JJ

result result =c

C. A logic error [2] 0

else: X

result =c

(vii) You have been asked to test the function `Funky(x,y,z)` using either an **exhaustive testing** or a **equivalence classes** testing strategy. Which method is best? Explain your answer. [2] 2

The best method would be either an equivalence classes testing strategy because this is much quicker than exhaustive testing JJ

(viii) Does this list of function calls

```
Funky(7,5,1)
Funky(5,5,9)
```

constitute **statement coverage** testing of function `Funky(x,y,z)`? **Explain** your answer. [2] 2

No this list of function calls doesn't constitute of statement coverage because all the values within the list are the same JJ

6

(ix) Write down a **minimal set** of calls to the function `Funky(x,y,z)` that will constitute a complete **path test** of this function. [2]

Funky( ) X
Funky( )

University of Cape Town ~~ Department of Computer Science

Computer Science 1015F ~~ 2017

Class Test 2

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 :

HEr HNDARB007

Name (optional) :

Abdoraghmaan

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	☐	☐	☐	☐	☐	☐				

Marks : 35

Time : 40 minutes

Instructions:

a) Answer all questions.

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

c) Show all calculations where applicable.

FOR OFFICIAL USE ONLY:

JB

Question	1	2	3	4	5	6	7	8
Max	10	7	6	12				
Marks								
Marker	7	3	5	8				

Question 1 - Functions [10]

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

(i) Implement the missing `prepare_alexa_string` function using string formatting and a docstring. The function should return a string that Alexa will eventually read out that says her favourite country is the `country` argument provided. For example, passing in the string "South Africa" should return the string "My favourite country is South Africa!". Likewise, calling `prepare_alexa_string ("Zimbabwe")` should return the string "My favourite county is Zimbabwe!" [4] 2

```
def prepare_alexa_str (country):
    """Function returns a string with Alexa's favourite country"""
    a = "My favourite country is "
    b = "!"
    return a + (country) + b
```

(ii) Assume that the return value of the function `alexa_fns.get_fav_country()` is "South Africa" and you have correctly implemented the function above. What is the output when the code from the `Q1.py` module is executed? [2] 2

DEBUG | South Africa | 8

4

(iii) What is the purpose of the following line? [2] 2

```
if __name__ == "__main__":
```

This executes main ~~because~~ function and allows to programmer to import this ~~program~~ program ✓✓

(iv) Rewrite the `main()` method to be a single line without changing the behaviour. [1] ✓1

```
print("DEBUG", alexa_fns.get_fav_country(), get_fav_number(3,5,7), sep="|")
```

(v) Give one reason why **global variables** are a bad idea. [1] 0

It cannot be changed. ✗

3

Question 2 - Hardware and Software [7]

(i) In the early 1800s, the Jacquard Loom used punch cards to control a loom in order to simplify the process of manufacturing complex designs for textiles. While this is not a general purpose computer, it is an example of an early computing device. The Jacquard system contains the equivalent of a CPU and Machine Code. Identify which part of the Jacquard loom corresponds to each of these. [2] 0

Punch cards

Software X

(ii) Give **two differences** between high-level and low-level languages. [2] 1

High level languages - They are easier for human to understand

Low level languages - They are easier for computers to understand

(iii) Von Neumann's Architecture described memory, a control unit, and what other elements? [1] 1

Memory - Stores all data and instructions in its memory

Control unit - This stores and executes instructions

ALU - This does all the calculations

Accumulator - Stores internal storage of ALU that stores data

Input - Input process that stores the given data into the machine

Output - Output obtains the result of the machine.

(iv) Give **one advantage** and **one disadvantage** of **assembler**. [2] 1

- Advantage - Very fast execution of program

Disadvantage - Program is for too long

3

Question 3 - String Problem Solving [6] 5

Write a function that takes in a string as an argument and switches the case of each alphabetic character. Your method should replace all of the lowercase letters in the original string with uppercase and replace all uppercase letters in the original string with lowercase. Other characters should remain unchanged.

You may use other string functions, such as `upper()` and `lower()`, character-based processing and/or string slicing as needed.

Your function will take a single argument: the original source string.

```
def convert_case (source):
    For i in range. Word = Source
    a = input("Enter
    max = len (source)
    for i in range (max):
        if source[i] == source.lower() source[i].lower():
            print (source.upper(), end="")
        else
            print(source.lower(), end="")

    max = len (source)
    For i in range (max):

        if source[i] = source[i].lower():
            print (source[i].upper(), end="")

        else if source[i] = source[i].upper():
            print (source[i].lower(), end="")

        elif source[i] = source[i].upper:
            print (source[i].lower(), end="")

        elif source[i] != source[i].upper or source[i]!=
            print (source[i], end="")
```

source[i].upper lower

Source

(Return a string, not print)

5

Question 4 – Testing [12]

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

(vi) **From the MK.py module**, give an example of:

A. A syntax error [2] 0

~~print(float(input("Enter your age!"))~~ X
~~but~~ return a C

B. A runtime error [2] 2

~~Division by zero~~ x=a ~~return c~~. √√
y=15/x There is no identifier
~~Division by zero~~ named C.

C. A logic error [2] 1

~~x=5~~ ~~Two if statements~~
~~y=10~~ Nested if statements is done
~~print((x+y))~~ ← ~~multiplication~~ incorrectly √

(vii) You have been asked to test the function `Funky(x,y,z)` using either an **exhaustive testing** or a **equivalence classes** testing strategy. Which method is best? Explain your answer. [2] 2

Equivalence classes, because it covers ~~major~~ all
data inputs in a much faster and simpler way.
√√

(viii) Does this list of function calls

```
Funky(7,5,1)
Funky(5,5,9)
```

constitute **statement coverage** testing of function `Funky(x,y,z)`? **Explain** your answer. [2] 2

No, because it doesn't executes when z > y.
This only covers if x > y and x == y √

7

(ix) Write down a **minimal set** of calls to the function `Funky(x,y,z)` that will constitute a complete **path test** of this function. [2]

(7, 5, 1) ✓
(5, 5, 9)
(7, 5, 6)

University of Cape Town ~~ Department of Computer Science

Computer Science 1015F ~~ 2017

Class Test 2

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 :

RDDVIVC01

Name (optional) :

Viveshan

Marks : 35

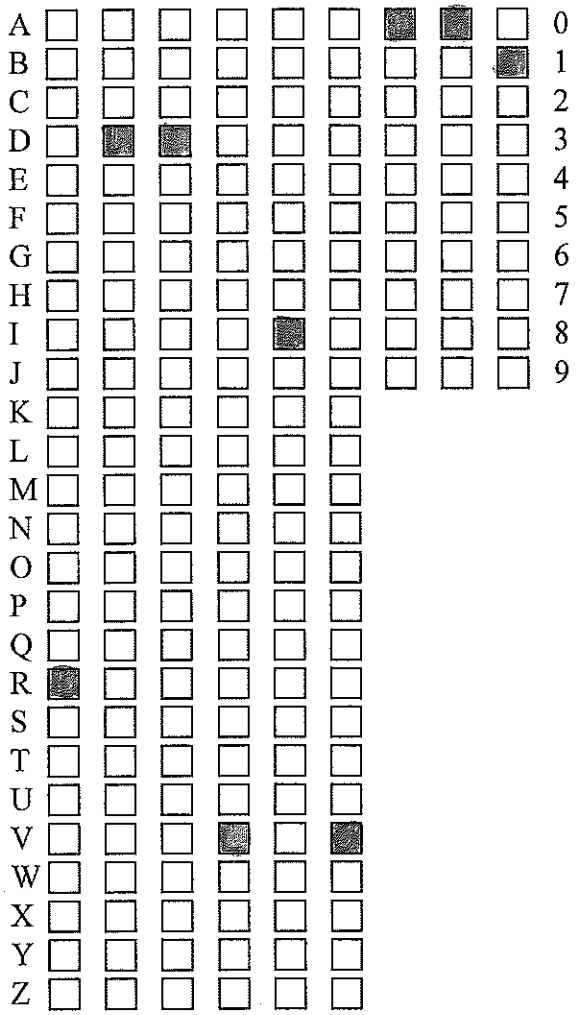
Time : 40 minutes

Instructions:

a) Answer all questions.

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

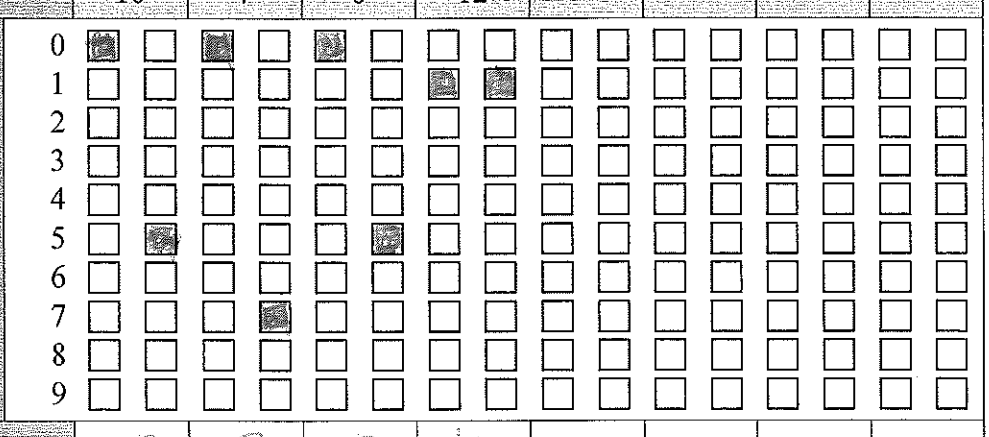
c) Show all calculations where applicable.



FOR OFFICIAL USE ONLY:

JB

Question	1	2	3	4	5	6	7	8
Max	10	7	6	12				
Marks								
Marker	5	7	5	11				



Question 1 - Functions [10]

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

(i) Implement the missing `prepare_alexa_string` function using string formatting and a docstring. The function should return a string that Alexa will eventually read out that says her favourite country is the `country` argument provided. For example, passing in the string "South Africa" should return the string "My favourite country is South Africa!". Likewise, calling `prepare_alexa_string ("Zimbabwe")` should return the string "My favourite county is Zimbabwe!" [4]

```
def prepare_alexa_str (country):
    """ Alexa favourite country """
    print ("My favourite county", x)

    return
```

(ii) Assume that the return value of the function `alexa_fns.get_fav_country()` is "South Africa" and you have correctly implemented the function above. What is the output when the code from the `Q1.py` module is executed? [2]

DEBUG | 'South Africa' | 8

(iii) What is the purpose of the following line? [2] 2

```
if __name__ == "__main__":
```

To call a function with name within a program, so that it may execute. This helps when you have another program but need to execute a part of this program. Allows you to import your own functions. ✓

(iv) Rewrite the `main()` method to be a single line without changing the behaviour. [1] 0

return ( ) ✗

(v) Give one reason why **global variables** are a bad idea. [1] 0

Global variables are fixed variables and not subjected to change ✗

Question 2 - Hardware and Software [7]

(i) In the early 1800s, the Jacquard Loom used punch cards to control a loom in order to simplify the process of manufacturing complex designs for textiles. While this is not a general purpose computer, it is an example of an early computing device. The Jacquard system contains the equivalent of a CPU and Machine Code. Identify which part of the Jacquard loom corresponds to each of these. [2] 2

CPU - The Loom ✓

Machine code - Punch cards ✓

(ii) Give **two differences** between high-level and low-level languages. [2] 2

1) High-level languages is easier for humans to use to code compared to low-level language which is harder for humans to code. ✓

2) High level language takes longer to execute, compared to low-level language ✓

(iii) Von Neumann's Architecture described memory, a control unit, and what other elements? [1] 1

Inputs and outputs ✓

(iv) Give **one advantage** and **one disadvantage** of **assembler**. [2] 2

Advantage - Converts high ~~ter~~ level language into machine code so it makes it easier for humans to program ✓

Disadvantage - Takes longer to execute ✓

7

Question 3 - String Problem Solving [6] 5

Write a function that takes in a string as an argument and switches the case of each alphabetic character. Your method should replace all of the lowercase letters in the original string with uppercase and replace all uppercase letters in the original string with lowercase. Other characters should remain unchanged.

You may use other string functions, such as `upper()` and `lower()`, character-based processing and/or string slicing as needed.

Your function will take a single argument: the original source string.

```
def convert_case (source):
    for x in range (0, len (source )) :
        if x > ord (97) and x < ord (122):
            print ( source. upper (), end "")
        else :
            print ( source. lower (), end " ")
    return ( )
```

(return string, don't print out)

5

Question 4 – Testing [12]

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

(vi) **From the MK.py module**, give an example of:

A. A syntax error [2] 2

```
print (x y z)
```

B. A runtime error [2] 2

```
result = c
```

C. A logic error [2] 2

```
if x == y or y == z or z == x
```

(vii) You have been asked to test the function `Funky(x,y,z)` using either an **exhaustive testing** or a **equivalence classes** testing strategy. Which method is best? Explain your answer. [2] 2

Equivalence classes texting ~~s~~tategy - Because exhaustive testing requires ~~infinitly~~ infinate number of inputs. ~~and~~

(viii) Does this list of function calls

```
Funky(7,5,1)
Funky(5,5,9)
```

constitute **statement coverage** testing of function `Funky(x,y,z)` ? **Explain** your answer. [2] 2

Yes - ~~Both~~ This list of functions tests for every statement. ~~The first one tested if~~

10

(ix) Write down a **minimal set** of calls to the function `Funky(x,y,z)` that will constitute a complete **path test** of this function. [2]

Funky (9, 6, 5)

Funky (9, 6, 7)

Funky (7, 7, 7)

University of Cape Town ~~ Department of Computer Science

Computer Science 1015F ~~ 2017

Class Test 2

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 :

RCHYATOOI

Name (optional) :

Yathin.

Marks : 35

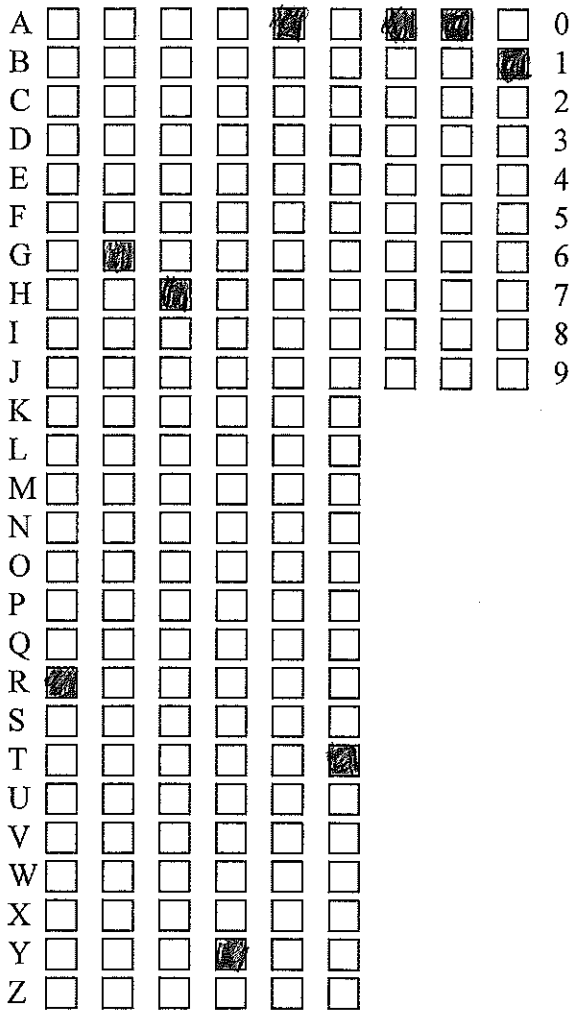
Time : 40 minutes

Instructions:

a) Answer all questions.

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

c) Show all calculations where applicable.



FOR OFFICIAL USE ONLY:

JB

Question	1	2	3	4	5	6	7	8
Max	10	7	6	12				
Marks	4	7	1	9				
Marker								

Question 1 - Functions [10]

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

(i) Implement the missing `prepare_alexa_string` function using string formatting and a docstring. The function should return a string that Alexa will eventually read out that says her favourite country is the `country` argument provided. For example, passing in the string "South Africa" should return the string "My favourite country is South Africa!". Likewise, calling `prepare_alexa_string ("Zimbabwe")` should return the string "My favourite county is Zimbabwe!" [4]

```
def prepare_alexa_str (country):
def prepare_alexa_str (country):
    """ get_fav_country """
    country = input ("Enter favourite country")
    Print ("My favourite country is ", + country, + "!.")

def get_fav_country
    """ get-fav-country """
    country = input ("Enter favourite country")
    Print ("My favourite country is ", + country + "!")
```

(ii) Assume that the return value of the function `alexa_fns.get_fav_country()` is "South Africa" and you have correctly implemented the function above. What is the output when the code from the `Q1.py` module is executed? [2]

```
South Africa. Debug.
DEBUG | South Africa | 8
```

(iii) What is the purpose of the following line? [2]

```
if __name__ == "__main__":
```

~~This line calls it makes~~ This line implies or checks if the __name__ variable is the same as the "__main__": variable. X

(iv) Rewrite the `main()` method to be a single line without changing the behaviour. [1]

__name__ == "__main__" : X

(v) Give one reason why **global variables** are a bad idea. [1]

They remain ~~true throughout~~ throughout an entire ~~program~~ program X

Question 2 - Hardware and Software [7]

(i) In the early 1800s, the Jacquard Loom used punch cards to control a loom in order to simplify the process of manufacturing complex designs for textiles. While this is not a general purpose computer, it is an example of an early computing device. The Jacquard system contains the equivalent of a CPU and Machine Code. Identify which part of the Jacquard loom corresponds to each of these. [2] 2

Cpu - the physical parts of the loom ✓

Machine code - punch cards ✓

(ii) Give **two differences** between high-level and low-level languages. [2] 2

- High level languages are easy to understand by humans whilst low level languages are easily understood by machines ✓
- High level languages need to be interpreted or compiled so that machines can understand it whilst ✓ low level languages do not need to be interpreted or compiled

(iii) Von Neumann's Architecture described memory, a control unit, and what other elements? [1]

- arithmetic logic unit ✓ (accumulator).
- input elements ✓
- output elements

(iv) Give **one advantage** and **one disadvantage** of **assembler**. [2] 2

- advantage: assembler is low level language and needs minimal translation time hence executes faster and is easily understood by machine ✓

disadvantage: ✓

- Harder for humans to understand and usually takes longer to program.

7

Question 3 - String Problem Solving [6]

Write a function that takes in a string as an argument and switches the case of each alphabetic character. Your method should replace all of the lowercase letters in the original string with uppercase and replace all uppercase letters in the original string with lowercase. Other characters should remain unchanged.

You may use other string functions, such as `upper()` and `lower()`, character-based processing and/or string slicing as needed.

Your function will take a single argument: the original source string.

```
def convert_case (source):
```

```
def get_lower_to_upper ():
    get_str = input ("Enter a sentence")      #Enter a sentence
    if ord('0') - get_str.ord('z')            #convert lower to upper char
    get_str = get_str.upper()
    return()

def get_upper_to_lower ():
    get_str = get_str.lower()
    ord.

    return ()
```

Question 4 – Testing [12]

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

(vi) **From the MK.py module**, give an example of:

A. A syntax error [2] 2

result = c is not defined.
Print (x y z) √√
(insert commas to fix)

B. A runtime error [2] 2

result = c result = c √√
(result = c is, whereby c is not defined)
If x == y or y == z or z == x.

C. A logic error [2] 0

result = c X
(result = c → whereby c is not defined)

(vii) You have been asked to test the function `Funky(x,y,z)` using either an **exhaustive testing** or a **equivalence classes** testing strategy. Which method is best? Explain your answer. [2] 2

equivalence classes testing. √√
- Exhaustive testing would require infinitely many inputs. which is impossible

(viii) Does this list of function calls

```
Funky(7,5,1)
Funky(5,5,9)
```

constitute **statement coverage** testing of function `Funky(x,y,z)`? **Explain** your answer. [2] 2

Yes. all paths of the function is tested at least once. √√

8

(ix) Write down a **minimal set** of calls to the function `Funky(x,y,z)` that will constitute a complete **path test** of this function. [2]

(7, 5, 1)

(5, 5, 9)

University of Cape Town ~~ Department of Computer Science

Computer Science 1015F ~~ 2017

Class Test 2

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 :

BHRTAA001

Name (optional) :

Taahir

Marks	**: 35**
Time	**: 40 minutes**

Instructions:

a) Answer all questions.

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

c) Show all calculations where applicable.

	0	1	2	3	4	5	6	7	8	9
A					■	■	■	■		
B	■								■	
C										
D										
E										
F										
G										
H		■								
I										
J										
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	10	7	6	12				
Marks								
Marker	6	2	2	9				

JB

Question 1 - Functions [10]

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

(i) Implement the missing `prepare_alexa_string` function using string formatting and a docstring. The function should return a string that Alexa will eventually read out that says her favourite country is the `country` argument provided. For example, passing in the string "South Africa" should return the string "My favourite country is South Africa!". Likewise, calling `prepare_alexa_string ("Zimbabwe")` should return the string "My favourite county is Zimbabwe!" [4] 2

```
def prepare_alexa_str (country):
```

```
~~Country = inp~~
Country = input("Enter favourite country: ")
return ("My favourite country is " + str(country),
        end = "!")
```

(ii) Assume that the return value of the function `alexa_fns.get_fav_country()` is "South Africa" and you have correctly implemented the function above. What is the output when the code from the `Q1.py` module is executed? [2] 2

DEBUG | South Africa | 8 ✓✓

4

(iii) What is the purpose of the following line? [2]

```
if __name__ == "__main__":
```

To ensure that the function defined as ; main() was actually ~~ran~~ correctly run as the main function when called ✓

(iv) Rewrite the `main()` method to be a single line without changing the behaviour. [1]

print("DEBUG"+ str(alert_fns.get_fav_country()) + str(get_fav_number(3517)), sep="|") ✓

(v) Give one reason why **global variables** are a bad idea. [1]

They ~~at~~ only exist within the defined function and thus cannot be accessed outside of it. ✓

Question 2 - Hardware and Software [7]

(i) In the early 1800s, the Jacquard Loom used punch cards to control a loom in order to simplify the process of manufacturing complex designs for textiles. While this is not a general purpose computer, it is an example of an early computing device. The Jacquard system contains the equivalent of a CPU and Machine Code. Identify which part of the Jacquard loom corresponds to each of these. [2] 2

The punch cards are the machine code and the loom is the CPU ✓✓

(ii) Give **two differences** between high-level and low-level languages. [2] 0

1. High level languages can accomplish harder tasks with less and more intuitive code ✗

2. High level languages are more stable and less likely to crash.

(iii) Von Neumann's Architecture described memory, a control unit, and what other elements? [1] 0

A Software, in the form of a coding language. ✗

(iv) Give **one advantage** and **one disadvantage** of **assembler**. [2] 1

Advantage - ~~it represents the capacity~~ It can work on old or outdated CPU's

Disadvantage - It is ~~along~~ a long method of writing code that is prone to many errors. ✓

Question 3 - String Problem Solving [6]

Write a function that takes in a string as an argument and switches the case of each alphabetic character. Your method should replace all of the lowercase letters in the original string with uppercase and replace all uppercase letters in the original string with lowercase. Other characters should remain unchanged.

You may use other string functions, such as `upper()` and `lower()`, character-based processing and/or string slicing as needed.

Your function will take a single argument: the original source string.

```
def convert_case (source):
    for i in range (len(source)):
        if source.type == str:
            if ord(source) >= 28:
                return(lower(source))
            elif ord(source) <= 27:
                return(upper(source))
```

Question 4 – Testing [12]

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

(vi) **From the MK.py module**, give an example of:

A. A syntax error [2] 2

line 4: print(x y z) . This is as there are no commas to separate the elements to be printed. √√

B. A runtime error [2] 2

"result = c" would not run as "result" looks like an undefined variable with the "=" present instead of brackets. √√

C. A logic error [2] 0

The values for (x, y, z) have been declared twice differently ~~after~~ after the function. ✗

(vii) You have been asked to test the function `Funky(x,y,z)` using either an **exhaustive testing** or a **equivalence classes** testing strategy. Which method is best? Explain your answer. [2] 2

equivalence, an exhaustive test would require one to test every possible value where as the same result could be achieved faster by testing the function using a equivalence class, as there are very few pathways. √√

(viii) Does this list of function calls

```
Funky(7,5,1)
Funky(5,5,9)
```

constitute **statement coverage** testing of function `Funky(x,y,z)`? **Explain** your answer. [2] 2

No, as not all pathways have been tested namely if x = y = z. √√

8

(ix) Write down a **minimal set** of calls to the function `Funky(x,y,z)` that will constitute a complete **path test** of this function. [2]

Funky(5,3,2) Funky(2,1,3)

Funky(5,5,5)