

University of Cape Town
Department of Computer Science

Computer Science CSC1010H

Class Test 2

Wednesday, 20 August 2014

Marks: 35

Time: 40 minutes

- Approximate marks per question are shown in brackets
- The use of calculators is permitted

NAME: Surname MOTABOLA Initials TIF
T1150

STUDENT NO: MTBT 11001 COURSE CODE: CSC 1010H

This paper consists of 6 questions and 6 pages (including this cover page).

Mark Allocation							
Question	Marks	Internal	External	Question	Marks	Internal	External
1	5			5	4		
2	7			6	6		
3	6						
4	7						
Total				Total			
Grand Total							
						Final Mark	
Internal Examiner:				External Examiner:			

Question 1. [5 marks]

Consider the following problem. Answer it appropriately.

The Petersens have recently moved to a new town and are arranging a surprise birthday party for their son Andre, and have invited three families from the neighbourhood, the Smiths, the Januarys and the Hectors. They plan to make up party packets for the kids to take home after the party, blue for boys and pink for girls.

Being super organised, Mrs Petersen with the help of Mr Petersen wants to determine how many of each colour party packet she needs to buy, and also how many of each colour she needs to put aside for each family.

They sit down and come up with the following information. Mrs Petersen remembers that the Hectors have a "pigeon pair", i.e. a boy and a girl. Mr Petersen recalls that the Januarys only have a set of identical twin boys. Mrs Petersen notes that she's only ever noticed two girls from these local families to come over to play. Mr Petersen notes that the Smiths have three children, since the family fits nicely into their family sedan when they go out.

You happen to be visiting the Petersens at this point, and want to impress them with the problem solving skills you've learnt at university. Using the information they've provided, determine how many of each colour party packet they need to buy and how many of each colour they need to allocate to each family and what the total number of party packets are.

Use a diagram to show how you solve the problem.

Use a diagram to show how you solve the problem.		No of color packs per family		
Petersens	$\begin{array}{l} \swarrow b \\ \searrow g \end{array}$	1 blue	1 pink	Smiths Januarys Hectors
Januarys	$\begin{array}{l} \swarrow b \\ \searrow b \end{array}$	2 blue		Blue - boys
Smiths	$\begin{array}{l} \swarrow g \\ \searrow b \\ \quad \searrow b \end{array}$	1 pink	2 blue	P - girls Key b for boy g for girl
Total number of packs		13	7	

[5]

Question 2. [7 marks]

Answer the following questions:

- a) When using debugging features in an IDE, what should the user typically do once execution has reached the breakpoint?

Resume execution by clicking on the ~~start~~ Continue debugging icon from the debug toolbar. then step over the program line after line. [2]

- b) When a new module has been defined, how do you ensure that it is accessible and can be imported into a program with no problems, i.e. "import newmodule" works?

use ~~--name=main--~~ ^{use} ~~--"name"--~~ = ~~+"main!"~~ - [1]

- c) Explain what happens in memory when Python makes successive recursive function calls.

All the results are stored and there is reference to the previous result to produce the next output. [1]

Indicate whether the following statements are True or False.

- d) The accepted Python coding convention for module names is long descriptive names in uppercase.

False [1]

- e) Curly brackets {} are used to enclose parameters to a function.

false [1]

- f) The print() function can be used to write to a file.

True [1]

Question 3. [6 marks]

Write a Python function called `draw_line()` which draws a horizontal line of characters. The `draw_line()` function should take two parameters, with the first being the size of the line (i.e. the number of characters) and the second parameter being the character with which to draw the line. This character parameter should have a default value of an asterisk (*).

Calling the `draw_line()` function with the following parameters should produce the corresponding output:

<code>draw_line(5)</code>	<i>produces</i>	*****
<code>draw_line(6, '\$')</code>		\$\$\$\$\$\$

size = int(input("Enter size"))
def draw_line(size, character = '):*
*pattern = size * character*
return pattern

draw_line(size, character)

[6]

Question 4. [7 marks]

Consider the following recursive function definition:-

```
def do_this(stuff):
    if len(stuff) == 0:
        return "" empty string
    else:
        return str(stuff[0] * 2) + do_this(stuff[1:]) end
```

[1, 2, 3] *[] [1] [2] [3]*

a) What datatype can the parameter to this function be?

int

[2]

b) What is the base case for this function?

0

[1]

c) Based on the `do_this()` function definition, what will the following statements display?

i. `print(do_this([1,2,3]))`

[1][1][2][3]

[2]

ii. `print(do_this("123"))`

1123

[2]

Question 5. [4 marks]

Consider the following Python program and answer the questions below:

```
def main():
    f = open('to_do_list.txt', 'a')
    while True:
        thing_to_do = input('Enter thing to do:')
        if thing_to_do == 'done': break
        else:
            f.write(thing_to_do + '\n')
    f.close()
```

`main()`

a) What is the name of the file created?

to-do-list

[1]

b) What mode is the file created in?

'a' append mode

[1]

c) Looking at the code, how does the user terminate the program?

By typing in 'done'

[1]

d) How will the information that the user enters be written in the file?

Information will be added or appended at the bottom of the already existing information with every input in its own line.

[1]

Question 6. [6 marks]

Consider the following definition of the classify_weight() function. Specify test cases which thoroughly test the function, using equivalence classes and boundary value. For each test case, specify whether it is an equivalence class value or a boundary value.

```
# classifies weight in kgs ✓
def classify_weight(w):
    if 0 < w <= 60:
        return "light"
    elif 60 < w <= 120:
        return "heavy"
    else:
        return "error"
```

	expected result
On Boundary values : 60	light
120	120
0	
equivalence classes	
2, 56 →	light
61, 120 →	heavy
123, 140 →	error
0 →	erroneous

[6]

University of Cape Town
Department of Computer Science

Computer Science CSC1010H

Class Test 2

Wednesday, 20 August 2014

Marks: 35

Time: 40 minutes

- Approximate marks per question are shown in brackets
- The use of calculators is permitted

NAME: Surname Initials
V
Sarugaser

STUDENT NO: SRGVARCOO1 COURSE CODE: CSC 1010H

This paper consists of 6 questions and 6 pages (including this cover page).

Mark Allocation							
Question	Marks	Internal	External	Question	Marks	Internal	External
1	5			5	4		
2	7			6	6		
3	6						
4	7						
Total				Total			
Grand Total							
						Final Mark	
Internal Examiner:				External Examiner:			

Question 1. [5 marks]

Consider the following problem. Answer it appropriately.

The Petersens have recently moved to a new town and are arranging a surprise birthday party for their son Andre, and have invited three families from the neighbourhood, the Smiths, the Januarys and the Hectors. They plan to make up party packets for the kids to take home after the party, blue for boys and pink for girls.

Being super organised, Mrs Petersen with the help of Mr Petersen wants to determine how many of each colour party packet she needs to buy, and also how many of each colour she needs to put aside for each family.

They sit down and come up with the following information. Mrs Petersen remembers that the Hectors have a "pigeon pair", i.e. a boy and a girl. Mr Petersen recalls that the Januarys only have a set of identical twin boys. Mrs Petersen notes that she's only ever noticed two girls from these local families to come over to play. Mr Petersen notes that the Smiths have three children, since the family fits nicely into their family sedan when they go out.

You happen to be visiting the Petersens at this point, and want to impress them with the problem solving skills you've learnt at university. Using the information they've provided, determine how many of each colour party packet they need to buy and how many of each colour they need to allocate to each family and what the total number of party packets are.

Use a diagram to show how you solve the problem.

	Blue (Boys)	Pink (Girls)
Smiths (3)	2	1
Januarys	2	-
Hectors	1	1
	5	2

∴ The Smiths with 3 children have 2 boys and 1 girl.

The Januarys have 2 boys

The Hectors have a boy and a girl.

∴ the Smith Petersens should buy 5 blue party packets for the boys and 2 pink party packets for the girls.

[5]

Question 2. [7 marks]

Answer the following questions:

- a) When using debugging features in an IDE, what should the user typically do once execution has reached the breakpoint?

Set over the following lines of code and
watch how values change.
[2]

- b) When a new module has been defined, how do you ensure that it is accessible and can be imported into a program with no problems, i.e. "import newmodule" works?

from newmodule() import * [1]

- c) Explain what happens in memory when Python makes successive recursive function calls.

It stores values and changes values each time it
runs or until it completes the function. [1]

Indicate whether the following statements are True or False.

- d) The accepted Python coding convention for module names is long descriptive names in uppercase.

False. [1]

- e) Curly brackets {} are used to enclose parameters to a function.

False. [1]

- f) The print() function can be used to write to a file.

False. [1]

Question 3. [6 marks]

Write a Python function called `draw_line()` which draws a horizontal line of characters. The `draw_line()` function should take two parameters, with the first being the size of the line (i.e. the number of characters) and the second parameter being the character with which to draw the line. This character parameter should have a default value of an asterisk (*).

Calling the `draw_line()` function with the following parameters should produce the corresponding output:

```
draw_line(5)           produces      *****
draw_line(6, '$')      produces      $$$$$$
```

```
def draw_line(length, character):
    character = '*'
    print(character * length)
    return str(character * length)
```

```
draw_line(5)
```

[6]

Question 4. [7 marks]

Consider the following recursive function definition:

```
def do_this(stuff):
    if len(stuff) == 0:
        return ""
    else:
        return str(stuff[0] * 2) + do_this(stuff[1:])
```

a) What datatype can the parameter to this function be?

```
int
```

[2]

b) What is the base case for this function?

```
0 length = 0 / len(stuff) == 0
```

[1]

c) Based on the `do_this()` function definition, what will the following statements display?

i. `print(do_this([1,2,3]))`

22 223 {2,2,3} 2,2,3 [2]

ii. `print(do_this("123"))`

223 [2]

Question 5. [4 marks]

Consider the following Python program and answer the questions below:

```
def main():
    f = open('to_do_list.txt', 'a')
    while True:
        thing_to_do = input('Enter thing to do:')
        if thing_to_do == 'done': break
        else:
            f.write(thing_to_do + '\n')
    f.close()
```

`main()`

a) What is the name of the file created?

to_do_list [1]

b) What mode is the file created in?

appending - 'a' to a textfile(txt) [1]

c) Looking at the code, how does the user terminate the program?

By entering the word 'done' [1]

d) How will the information that the user enters be written in the file?

a word on each line or a task on each line. [1]

Question 6. [6 marks]

Consider the following definition of the *classify_weight()* function. Specify test cases which thoroughly test the function, using equivalence classes and boundary value. For each test case specify whether it is an equivalence class value or a boundary value.

```
# classifies weight in kgs
def classify_weight(w):
    if 0 < w <= 60:
        return "light"
    elif 60 < w <= 120:
        return "heavy"
    else:
        return "error"
```

Boundary values : 0, 60, 120

~~0 = 'error'~~ 60 = light 120 = heavy.

Equivalence values : 52, 75, 109

52 = light 75 = heavy 109 = heavy

[6]

University of Cape Town
Department of Computer Science

Computer Science CSC1010H

Class Test 2

Wednesday, 20 August 2014

Marks: 35

Time: 40 minutes

- Approximate marks per question are shown in brackets
- The use of calculators is permitted

NAME: Surname Initials
SEWNAIRAIN S

STUDENT NO: SWNSHA007 COURSE CODE: CSC 1010H

This paper consists of 6 questions and 6 pages (including this cover page).

Mark Allocation							
Question	Marks	Internal	External	Question	Marks	Internal	External
1	5			5	4		
2	7			6	6		
3	6						
4	7						
Total				Total			
Grand Total							
						Final Mark	
Internal Examiner:				External Examiner:			

Question 1. [5 marks]

Consider the following problem. Answer it appropriately.

The Petersens have recently moved to a new town and are arranging a surprise birthday party for their son Andre, and have invited three families from the neighbourhood, the Smiths, the Januarys and the Hectors. They plan to make up party packets for the kids to take home after the party, blue for boys and pink for girls.

Being super organised, Mrs Petersen with the help of Mr Petersen wants to determine how many of each colour party packet she needs to buy, and also how many of each colour she needs to put aside for each family.

They sit down and come up with the following information. Mrs Petersen remembers that the Hectors have a "pigeon pair", i.e. a boy and a girl. Mr Petersen recalls that the Januarys only have a set of identical twin boys. Mrs Petersen notes that she's only ever noticed two girls from these local families to come over to play. Mr Petersen notes that the Smiths have three children, since the family fits nicely into their family sedan when they go out.

You happen to be visiting the Petersens at this point, and want to impress them with the problem solving skills you've learnt at university. Using the information they've provided, determine how many of each colour party packet they need to buy and how many of each colour they need to allocate to each family and what the total number of party packets are.

Use a diagram to show how you solve the problem.

Use a diagram to show how you solve the problem.

	J	H	S	
B	✓✓	✓	✓✓✓	
P		✓	✓	2 girls
				2 2 Blues & 1 Pink for Smiths
				2 2 Blues & 0 Pink for January's
				2 1 Blue & 1 Pink for Hector's.

	S	J	H	Total	
Blue/Boys	✓✓	✓✓	✓	5	2 Needs to buy 2 Pink
Pink/Girls	✓	×	✓	2	Party Pack, 5 Blue
Total	3	2	2	7	Party Packs
					2 Total number

of Party Packs are 7

[5]

[5]

Question 2. [7 marks]

Answer the following questions:

- a) When using debugging features in an IDE, what should the user typically do once execution has reached the breakpoint?

Once a breakpoint is set it will halt execution ∴ start debugging
do user input/output in Debug I/O window, and i.e. Start execution
watch how the variable change in stack data in debug mode.
window ∴ stop debugging when error found. [2]

- b) When a new module has been defined, how do you ensure that it is accessible and can be imported into a program with no problems, i.e. "import newmodule" works?

* That it is saved in the same directory / file path. * You Import the module name only.
* & when using it, its module name. function. [1]

- c) Explain what happens in memory when Python makes successive recursive function calls.

So when every time a function is called it is placed / stored
in the Run Time Stack and only removed when completely evaluated. [1]

Indicate whether the following statements are True or False.

- d) The accepted Python coding convention for module names is long descriptive names in uppercase.

False. [1]

- e) Curly brackets {} are used to enclose parameters to a function.

False [1]

- f) The print() function can be used to write to a file.

True. [1]

Question 3.**[6 marks]**

Write a Python function called `draw_line()` which draws a horizontal line of characters. The `draw_line()` function should take two parameters, with the first being the size of the line (i.e. the number of characters) and the second parameter being the character with which to draw the line. This character parameter should have a default value of an asterisk ('*').

Calling the `draw_line()` function with the following parameters should produce the corresponding output:

<code>draw_line(5)</code>	<i>produces</i>	*****
<code>draw_line(6, '\$')</code>		\$\$\$\$\$\$

```
def draw_line(size, character="*"):
    for i in range(size):
        return size * character
def main():
    print(draw_line(5))
    print(draw_line(6, "$"))
main
```

[6]

Question 4.**[7 marks]**

Consider the following recursive function definition:

```
def do_this(stuff):
    if len(stuff) == 0:
        return ""
    else:
        return str(stuff[0] * 2) + do_this(stuff[1:])
```

a) What datatype can the parameter to this function be?

~~String~~ * Integers.
*: List

[2]

b) What is the base case for this function?

If the length of stuff is nothing that is 0/zero i.e. its empty [1]
3. return an empty string. 3. stopping condition if `len(stuff) == 0:`
return ""

c) Based on the `do_this()` function definition, what will the following statements display?

i. `print(do_this([1,2,3]))`

2 4 6

[2]

ii. `print(do_this("123"))`

11 22 33

[2]

Question 5. [4 marks]

Consider the following Python program and answer the questions below:

```
def main():  
    f = open('to_do_list.txt', 'a')  
    while True:  
        thing_to_do = input('Enter thing to do:')  
        if thing_to_do == 'done': break  
        else:  
            f.write(thing_to_do + '\n')  
    f.close()  
  
main()
```

a) What is the name of the file created?

to-do-list.txt

[1]

b) What mode is the file created in?

appending / append mode

[1]

c) Looking at the code, how does the user terminate the program?

If the user enters done then it will break & terminate 3 step.

[1]

d) How will the information that the user enters be written in the file?

By using the write() method, all the things to do will be written into the file each being on a new line.

[1]

Question 6. [6 marks]

Consider the following definition of the *classify_weight()* function. Specify test cases which thoroughly test the function, using equivalence classes and boundary value. For each test case specify whether it is an equivalence class value or a boundary value.

```
# classifies weight in kgs
def classify_weight(w):
    if 0 < w <= 60:
        return "light"
    elif 60 < w <= 120:
        return "heavy"
    else:
        return "error"
```

Equivalence Class - 45	} Category Values
Equivalence Class - 75	
Equivalence Class - -1	} Erroneous Values.
Equivalence Class - 121	

On Boundary Value - 0	Boundary Value above = 1, 61, 121
On Boundary Value - 60	Boundary Value below = -1, 59, 119
On Boundary Value - 120	

[6]