





# Unit 2 – ICS4U0 – Methods, Strings and Recursion

Sample Test – October 11, 2022

Name: Solution

Total	%	Knowledge 	Communication 	Thinking 	Application 
(112)	%	(30)	(29)	(23)	(30)

## Knowledge

1. Looking at this String, what would be output after each statement: `String g = "Gorilla";`

/12

0	1	2	3	4	5	6
G	O	R	I	L	L	A

- (a) //Fill in the memory diagram
- (b) `output.setText (" " + g.length ());`
- (c) `output.setText (" " + g.compareTo ("zebra") > 0);`
- (d) `output.setText (" " + g.equals ("Gorilla"));`
- (e) `output.setText (" " + g.charAt (1));`
- (f) `output.setText (g.toUpperCase ());`
- (g) `output.setText (g.replace ('l', 'm'));`
- (h) `output.setText (g.substring (2, 5));` *stop before 5*
- (i) `output.setText (" " + g.indexOf ('r'));`
- (j) `output.setText (" " + (int) g.charAt (6));` *a=97, A=65*
- (k) `output.setText (g.substring (0, g.indexOf ('l')));`
- (l) `output.setText (g.substring (g.length () / 3, g.length ()));` *3 (cut off decimal) 7*

.....7.....  
 .....false.....  
 .....true.....  
 .....O.....  
 .....GORILLA.....  
 .....Gorimma.....  
 .....ril.....  
 .....2.....  
 .....97.....  
 .....Gori.....  
 .....illa.....

as 2.2  
String  
methods

Also 1. Names  
Code.

2. This is the original String: `String f = "Fishes swimming";` Match the code with the output.

/6

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
F	i	s	h	e	s		S	w	i	m	m	i	n	g

b	Fishes swimming	a. <code>result.setText (" " + f.length ());</code>
c	h	b. <code>result.setText (f);</code>
h	true	c. <code>result.setText (" " + f.charAt (3));</code>
j	FISHES SWIMMING	d. <code>result.setText (" " + f.equals ("Gorilla"));</code>
d	false	e. <code>result.setText (f.substring (5, 9));</code>
a	15	f. <code>result.setText (f.replace ('s', 'q'));</code>
i	Fish	g. <code>result.setText (f.replace ('i', 'v'));</code>
f	Fiqheq qwimming	h. <code>result.setText (" " + f.compareTo ("yaks") &lt; 0);</code>
e	s sw	i. <code>result.setText (f.substring (0, 4));</code>
g	Fvshes swmmvng	j. <code>result.setText (f.toUpperCase ());</code>

as 2.2  
String  
methods

3. Using the method signatures, call the methods.

/6

(a) Signature: `public void dog()`

Call:

`dog` ();

as 2.1  
method  
calling

(b) Signature: `public int clearImage(String word)`

Call:

`String` word = IO.inputString("Enter a word: ");  
`int` answer = `clearImage` (`word`);

(c) Signature: `public boolean win(int amt)`

Call:

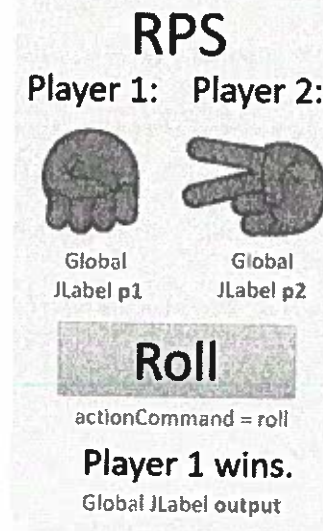
`int` amt = IO.input `Int` ("Enter a number: ");  
`boolean` answer = `win` (`amt`);

4. Create a method to change the rock, paper or scissors images. Then, call it in the onClick method.

/6

(a) Fill in the setPic method:

```
public void setpic(JLabel b, int num) {
    if (num == 0)
        b.setIcon(createImageIcon("rock.png"));
    else if (num == 1)
        b.setIcon(createImageIcon("paper.png"));
    else
        b.setIcon(createImageIcon("sciss.png"));
}
```



The picture names, the corresponding random numbers.



rock.png, 0



paper.png, 1



sciss.png, 2

(b) Fill in the onClick method, calling the setPic method for each player.

```
public void actionPerformed(ActionEvent e) {
    //random number for each
    int r1 = (int) Math.random()*3;
    int r2 = (int) Math.random()*3;
```

//two method calls:

```
setpic ( p1 , r1 );
setpic ( p2 , r2 );
```

as Pizza Party  
for Reusability in  
ORATE.

## Communication

5. Consider this recursive method.

as 2.4, 2.5 Recursion

/9

```
public int factorial (int n) {
    if (n == 1) // Base Case
        return 1;
    else
        return factorial (n - 1) * n; // Recursive Case
}
```

Identify the following:

(a)	The return type	int
(b)	The method name	factorial
(c)	The parameter name	n
(d)	The parameter type	int
(e)	Circle and label the base case and recursive case	w

Evaluate the following:

(f) factorial(1).

1

(g) factorial(4).

24

(h) factorial(0).

Error (not in base case)

6. Provide the phrase or number required.

/10

1 1 2 3 5
parameter
loop
charAt
infinite loop
stopping condition
Koch snowflake
method
Abstraction
Recursive

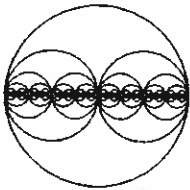
- (a) The first 5 terms of the Fibonacci sequence.
- (b) A variable that is passed into a method.
- (c) Every recursive method can be re-coded as this.
- (d) A String function that returns a char.
- (e) The equivalent of a stack overflow error on a loop.
- (f) The equivalent of a base case on a loop.
- (g) A fractal with a finite perimeter and infinite area.
- (h) A sub-program or function in java.
- (i) The 'A' from the ORATE methods mnemonic.
- (j) A method that calls itself.

Many places.

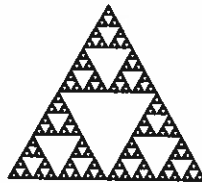
7. Match the fractal with its name:

/6

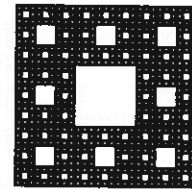
A.



B.



C.



D.



E.



F.



as 2.7, 2.8  
Fractals

Write the match below. Note that some names have no matching picture.

A	Cantor's Cheese	E	Cantor's set		Fern fractal
	Koch snowflake	P	Julia Set	F	Mandelbrot set
C	Sierpinski's carpet	B	Sierpinski's gasket		Lorenz attractor

8. Name two professions that would program fractals in real life. Explain why. (2 sentences)

/2

① A stock market investor might want to simulate overall stock market trends using the Lorenz attractor to help them make predictions.

② A CGI coder making a movie might use fractals such as the fern, tree, or texture fractals to make realistic plants in the background of a fantasy movie like Avatar.

9. Why is recursion useful? (2 points, one sentence each)

/2

① It can be more efficient than loops. For example, when sorting, Quicksort uses recursion to gain its speed.

② It can naturally suit the situation being coded. The Fibonacci sequence is self-referential - term(n) is term(n-1) + term(n-2). This definition is in terms of itself and thus is suited to recursive code.

# Thinking

10. Trace the following method with the method call `result.setText("" + tree(5));`

/5

```
public int tree (int n) {
    if (n == 1 || n == 0)
        return 1;
    else
        return (n) + tree (n - 2);
}
```

2.5 Recursion

tree(0) = 1	tree(1) = 1	tree(2) = 2 + tree(0) = 2 + 1 = 3	tree(3) = 3 + tree(1) = 3 + 1 = 4	tree(4) = 4 + tree(2) = 4 + 3 = 7	tree(5) = 5 + tree(3) = 5 + 4 = 9
----------------	----------------	--	--	--	--

11. Write a recursive method for the following sequence: 1 3 6 10 15 21 28 36 45  
For example, `seq(1)` returns 1 and `seq(3)` returns 6.

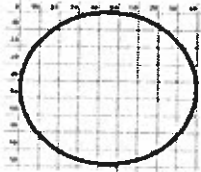
/5

```
public int seq (int n) {
    if (n <= 1)
        return 1;
    else
        return seq(n-1) + n;
}
```

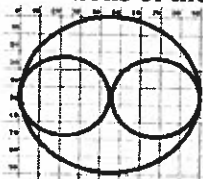
2.5 Recursion

5. Recursive Sequences Code

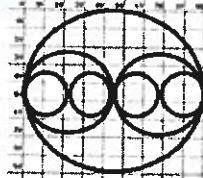
12. These are the first four generations of the Cantor's Cheese fractal.



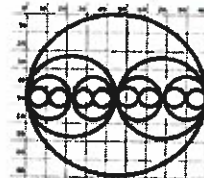
Generation 1



Generation 2



Generation 3



Generation 4

(a) How many circles are drawn in each generation of Cantor's Cheese?

/3

Generation	1	2	3	4	5	6	7	8	9
New Circles	1	2	4	8	16	32	64	128	256
Total Circles	1	3	7	15	31	63	127	255	511

(b) Write a recursive method to find the total number of circles after n generations.

/5

```
public int cheese (int n) {
    if (n <= 1)
        return 1;
    else
        return cheese(n-1) * 2 + 1;
}
```

5. Recursive Sequences Code



13. Circle and correct an error in each line of code.

/5

String s = "chipmunk";

answer.setText (" " + s.length());

answer.setText (s.charAt (1) + " " + s.charAt (4));

answer.setText (s.substring (3));

answer.setText (" " + s.compareTo ("squirrel"));

Strings

## Application

14. This is the original String: String m = "Moose Loose";

/8

0	1	2	3	4	5	6	7	8	9	10
M	o	o	s	e		L	o	o	s	e

As 4 Apples/Bananas Code

+ 2.2 Strings Sheet

Write the code to produce the following output:

(a)	MOOSE LOOSE	output.setText ( m.toUpperCase() );
(b)	Miise Liise	output.setText ( m.replace ('o', 'i') );
(c)	L	output.setText ( m.charAt (6) );
(d)	Moose	output.setText ( m.substring (0, 5) );
(e)	11 //the length	output.setText ( m.length() );
(f)	32 //ASCII of space	output.setText ( (int) m.charAt (5) );
(g)	6 //position of L	output.setText ( m.indexOf ('L') );
(h)	MooLoo	output.setText ( m.substring (0, 3) + m.substring (6, 9) );

15. Write a method named comeAndGo whose parameter is a String of any length. It returns true if first 2 chars in the String also appear at the end of the String and false otherwise.

Example 1: Input? abba Output? false	Example 2 Input? edited Output? true	Example 3 Input? ed Output? true	Example 4 Input? a Output? false	Example 5 Input? yy Output? true
--	--	--	--	--

public boolean comeAndGo (String s) {

if (s.length() < 2)  
return false;

else {

String firsthalf = s.substring (0, 2);

String lasthalf = s.substring (s.length()-2, s.length());

if (firsthalf.equals (lasthalf))

return true;

else

return false;

}

}

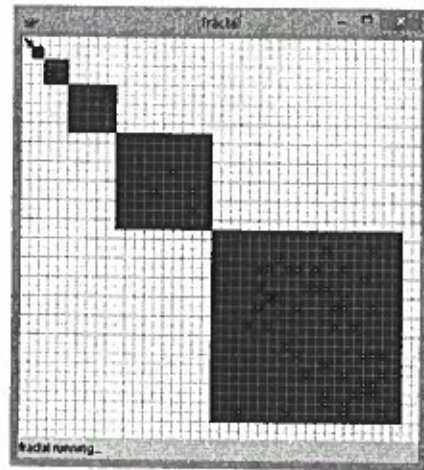
As 1. Names  
Code. No Loops.  
Substring. CharAt.

/7

16. Code this fractal.

These are the co-ordinates of the generations:

Gen	X	Y	Side length
1	200	200	200
2	100	100	100
3	50	50	50
4	25	25	25
5	12	12	12
6	6	6	6
7	3	3	3



```
import java.awt.*; import java.Applet.*;
public class fractal extends Applet {
    public void paint (Graphics g) {
        square (200);
    }
}
```

```
public void square (int n) {
```

method start word      return type (output)      method name (look above)      par type      par name (in)

```
    if (n < 3)
        return;
    else {
        Graphics g = getGraphics();
        g.fillRect (n, n, n, n);
        square (n/2);
    }
}
```

As 6 Fractals  
Code; also sheet 8  
Recursive Drawing

```
} } }
```

17. Use the string that the user has entered. It will have no capital letters – only small letters or numbers. Separated it into two Strings, one with numbers, one with small letters, which are printed on the screen.

Example 1:	Example 2:	Example 3	Example 4
Word? abc123	Word? H2134	Word? yyyy	Word? 1ttt23
abc	Hi	yyyy	ttt
123	234		123

```
String s = IO.inputString("Enter a string of numbers and letters: ");
s=s.toLowerCase();
```

```
String num = "";
String letter = "";
for (int i=0; i<s.length(); i++) {
    if (s.charAt(i) >= 'a' && s.charAt(i) <= 'z')
        letter += s.charAt(i);
    else
        num += s.charAt(i);
}
System.out.println (letter);
System.out.println (num);
```

As 2 Vote Count  
Code. Has a loop.  
uses charAt.