**Self-Assessment Quiz #4 (week 5):**

|  |
| --- |
| name: NICOLE CARRERA |

Variables and Loops: char, boolean and nested for loops

Functions: function calls, writing your own functions

**Only answer Q1 if you are NOT confident with a while loop and boolean tests** (these are used inside if statements and while loops), otherwise start at Q3

Q1:

int p=1;

while(p<16){

ellipse(p\*3,50,20,20);

p=p+4;

}

println("p is: "+p);

When p=1, is p<16 true or false? TRUE

What happens next?

What is drawn? **4 OVERLAPPING CIRCLES**

What is the next value of p? **5**

Then describe what happens for each value of p, including whether the statement tested inside the while () is true or false, what is drawn, what the next value of pi is…

TRUE WITHIN THE WHILE LOOP

* P is 1. P \* 3 = 3. Position X is 3
* P is 5. P \* 3 = 15. Position X is 15
* Then p is 9. P \* 3 = 45. Position X is 3
* Then p is 13. P \* 3 = 135. Position X is 3

FALSE WITHIN THE WHILE LOOP

The p is 17. NO CIRCLE IS DRAWN AS P >16

Q2: Nested for loop

What is printed out:

for(int j=0; j<2; j++){

for(int i=0; i<3; i++){

println("j is "+j+"and i is "+i);

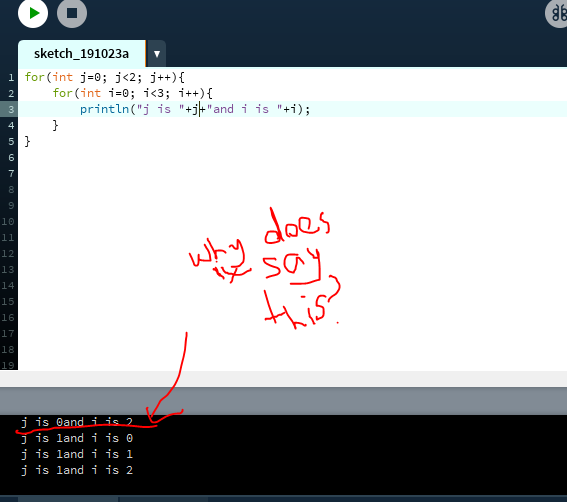
}

}

"j is "+0+"and i is "+0

"j is "+1+"and i is "+1

"j is "+1+"and i is "+2



Q3: Another nested for loop. Do not attempt to solve this immediately, and do not run this code. Instead think e.g. what happens when k=0 and j=0? what will be printed? Now what happens when k=0 and j=1 what will be printed, then k=0 and j=2 etc.

Note that println() will not print any characters but will make/force a new line break.

for (int k=0; k<2; k++) {

for (int j=0; j<4; j++) {

for (int i=0; i<j+1; i++) {

print("\*");

}//end of for i

println();

}//end of for j

}//end of for k

Draw what is printed out:

(Ignore the words)

GAPS

One \*

Two \*\*

Three \*\*\*

Four \*\*\*\*

One \*

Two \*\*

Three\*\*\*

Four\*\*\*\*

Q4: Functions are defined using a function signature – the return type, function name, brackets containing any parameters (type and name, separated by commas) – followed by the instructions/code inside curly brackets. This sets up the rules for what a function will do when it is called/used.

For example, the **function definition** for a rectangle (excluding the instructions) in Processing would probably look like:

void rect(float x,float y, float w, float h){…}

the instruction to **make/call** a rectangle is e.g.

rect(23,45,78,54);

Provide examples of function calls based on the following function definitions. You will need to make up argument values for each function call:

void createMonster(int height, int numberOfTeeth){…}

function call: **createMonster(20,100);**

void createTroll(float waistMeasurement, int numberOfWarts){…}

function call: **createTroll(36.32,12);**

void buyWeapon(float price, char status){…}

function call: **buyWeapon(100, ‘c’);**

void createHelper(boolean isAlive, char nameInitial, int wealth){…}

function call: **createHelper(true, ‘n’,’200);**

Q5: function definition & call

boolean isThisEven(){

int oe = int(random(20));

if(oe%2==0){

return true;

}

else{

return false;

}

}

Declare a variable called ‘isEven’ and show how it is used to store (take in) the return value when the isThisEven() function is called:

**isEven = isThisEven();**

Q6: function definition

int r1=(int)random(50);

int r2=(int)random(50);

void setup() {

int higher = getHigher(r1, r2);

println("the highest value is: "+higher);

}

//missing getHigher function definition

What is the range of values that r1 and r2 can take?

Between 0 and 50.

You need to write the getHigher function definition. This first prints out the values of r1 and r2 (for debugging purposes) then returns whichever is higher.

**int getHigher(int r1, int r2) {**

**print(r1);**

**print(r2);**

**if (r1 > r2) {**

**return r1;**

**} else {**

**return r2**

**}**

**}**

Q7: Which of the following initialisations of primitives will NOT compile:

boolean flag = “true”; **THIS ONE**

boolean flagb = false;

boolean flagc = ‘false’; **THIS ONE**

char initial = b;

char init = ‘c’; **THIS ONE**

char ini = ‘2’;

char in =”r”; **THIS ONE**

char i = ‘er’; **THIS ONE**

float f=2; **THIS ONE**

int w = 2;

int p = 3.4; **THIS ONE**

Q8: Fill in the gaps about functions using the terms from the table. Not all terms will be used:

|  |  |  |
| --- | --- | --- |
| definition | call/s | instruction |
| return type | void | argument |
| Parameter/s | scope | curly |
| Round/curved | variable |  |
| Define/s/d | Declare/s/d | cast |
| boolean | comma | float |

a) The draw function runs the instructions written inside **the curly\_\_\_\_\_\_\_** brackets by the developer. It does not return any value so its return type is \_\_ **void \_\_\_\_\_\_\_\_\_\_**

b) A variable that is \_ **called\_\_\_\_\_\_\_\_\_** inside a function is called a **\_local**\_\_\_\_\_\_\_\_ variable

c) The code below shows a function \_\_\_\_\_\_\_\_\_\_\_\_\_. The \_**return**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ for this function is an int. The variable inside the **\_curly\_\_\_\_\_\_\_\_\_** brackets in the function signature is called a \_**\_parameter**\_\_\_\_\_\_\_\_\_\_\_\_\_. Variable ‘r’ is a \_**defined\_\_\_\_\_\_\_\_\_\_** variable. When the function ends this variable is out of \_**scope\_\_\_\_\_\_\_\_\_\_\_** , i.e. no longer exists. The task of the Processing int(..) function is to \_**declare\_\_\_\_\_\_\_\_\_** the \_ **return type \_** returned by the random function into an int.

int checkDb(float code){

int r;

boolean checked = checkExists(code);

if(checked){

return int(random(r));

}

else

return -1;

}

d) The instruction “int c” \_**declares\_\_\_\_\_\_\_\_\_\_\_\_** a variable. This code \_**defines**\_\_\_\_\_\_\_\_\_\_\_\_\_ the checkDb function. The value inside the brackets is called the \_**parameter**\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

int c;

//missing code to read in a user’s value and store inside c

int myValue=checkDb(c);//uses the function in part c

e) This code will generate a compilation error as a function **\_\_void\_\_\_\_\_\_\_\_\_\_\_\_\_\_** is nested inside another function\_\_\_**void\_**\_\_\_\_\_\_\_\_ .

void draw person(){

drawHead();

void drawBody(){

fill(200,0,150);

rect(60,80,50,100);

}

}

Write here what you would have to do to correct this (and ensure the code draws a person, i.e. a body with a head):

void draw(){

size(1000,1000);

drawHead();

drawBody();

}

void drawBody() {

fill(200, 0, 150);

rect(60, 80, 50, 100);

}

void drawHead() {

fill(200, 0, 120);

ellipseMode(CENTER);

ellipse(60, 80, 50, 100);

}