



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



PREPARATORIA 9

Due date:
JUNE 07 2017

INFORMATION AND COMMUNICATION TECHNOLOGY II ACTIVITIES FINAL PORTFOLIO

DO THIS ON
YOUR OWN.

NAME: _____	GROUP #. _____	LIST # _____
-------------	----------------	--------------

IF ANSWERS CAN'T BE READ, THEY'LL COUNT AS WRONG.

STAGES 3 OR 4: KAREL THE ROBOT

LIST ARE KAREL'S BASIC COMMANDS (5 PTS)?

1.
2.
3.
4.
5.

WHAT ARE KAREL'S CONTROL STATEMENTS (NAME AND DEFINE BRIEFLY) (6 PTS):

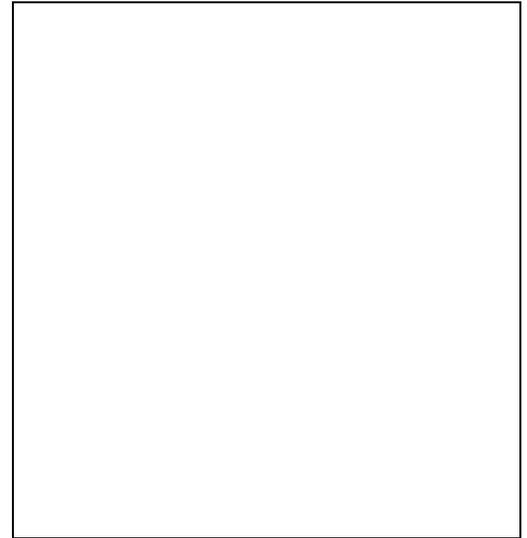
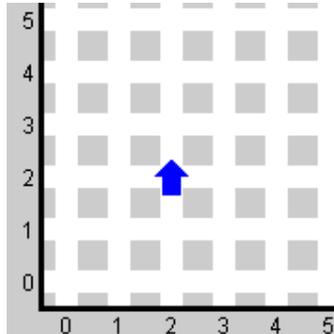
1.
2.
3.

<p>CONSIDERING ALL YOU HAVE LEARNED ABOUT KAREL'S CAPABILITIES (MOVE, TURN, HOLD, PICK AND DROP BEEPERS, DETERMINE IF THERE'S A WALL IN FRONT OR ON THE SIDE, NOTICE IF SHE'S STANDING OVER A BEEPER AND DETECT HER ORIENTATION), <u>MAKE A SKETCH</u> OF HER APPEARANCE. BE SURE TO INCLUDE AND LABEL ALL THE ELECTRONIC DEVICES SHE MUST HAVE. (5 PTS)</p>	
---	--

CODINGS – PART 1 (3 PTS)

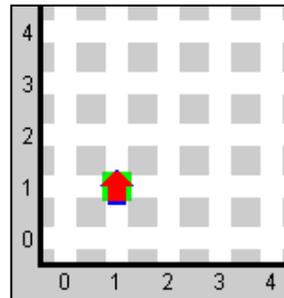
- 1) Use the image in the middle to draw the path that Karel will follow with the given code.
- 2) Write an alternate solution that will make Karel perform the same task as in the following code.

```
class program {
    program() {
        turnleft();
        while (notFacingNorth()) {
            move();
            if (frontIsClear()) {
                turnleft();
            }
        }
        move();
        turnoff();
    }
}
```

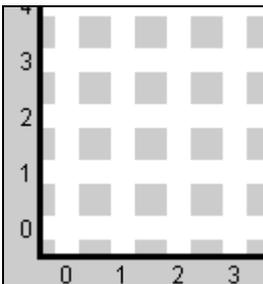


CODINGS – PART 2 (3 PTS)

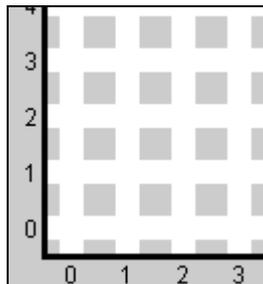
You've learned that brackets are mandatory in Java programming, particularly with control commands. Given the initial world below in which Karel holds as many beepers as necessary, describe the conditions of the final world as she follows the instructions in each of the three given codes. That is: (1) draw Karel in her final position and orientation (2) indicate the amount and location of the beepers.



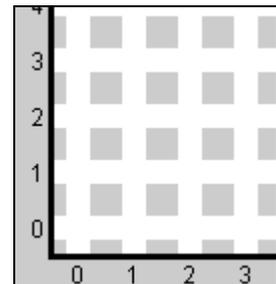
```
...
iterate (3) {
    putbeeper();
    move();
    turnleft()
}
...
```



```
...
iterate (3) {
    putbeeper();
    move();
    }
    turnleft();
}
...
```



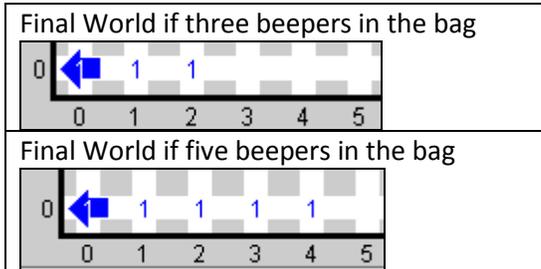
```
...
iterate (3) {
    putbeeper();
    }
    move();
    turnleft
}
...
```



CODINGS- PART 3 (4 PTS)

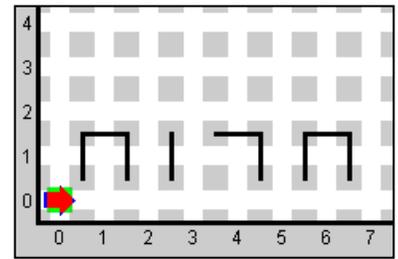
Karel is playing Hansel and Gretel. She will be dropping one beeper in every step she makes, coming back home when they're finished. She doesn't know how many beepers she has.

(Use your best handwriting.)



CODINGS- PART 6 (5 PTS)

Take as given the world shown in the figure. The situation is that karel wants to redecorate her home with some carpets. She's willing to place a rug only in the three sided rooms of her house. For doing so, karel must check each of the six rooms and place the rug only if the area is enclosed with three walls.

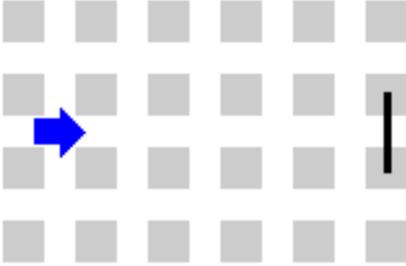
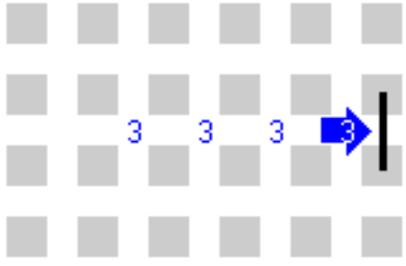


Checking the code below you will see that the programmer included modules. Write down the specific instructions for each module. (Use your best handwriting)

<pre> class program { program() { move(); iterate (6) { getIn(); CheckAndDrop(); getOut(); } turnBack(); goHome(); turnoff(); } } </pre>	<pre> void getIn() { } </pre>
	<pre> void CheckAndDrop() { } </pre>
	<pre> void getOut() { } </pre>
	<pre> void turnBack() { } </pre>
	<pre> void goHome() { } </pre>

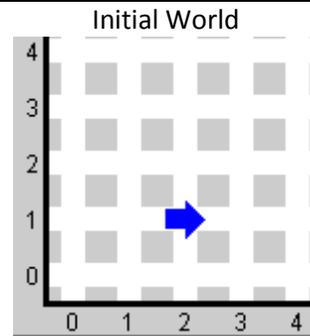
WORKING WITH PARAMETERS (3 PTS)

A parameter is a value that is transferred from one part of the program to another. Review the following example.

<p>Initial world</p> 	<p>Final world</p> 	<pre> class program { void dropbeeper(n) { iterate (n) { putbeeper(); } } program() { while (frontIsClear()) { move(); dropbeeper(3); } turnoff(); } } </pre>
--	--	---

Analyze:

Find the final world for each of the given codes (the options are at the bottom). Consider Karel has four beepers in her bag.



```

class program {
    void dropbeeper(n) {
        iterate (n) {
            turnleft();
        }
        iterate (n) {
            move();
        }
        putbeeper();
    }
    program() {
        while
        (anyBeepersInBeeperBag()) {
            dropbeeper(2);
        }
        turnoff();
    }
}
        
```

Answer

```

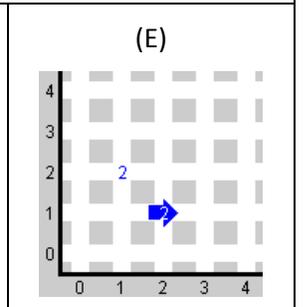
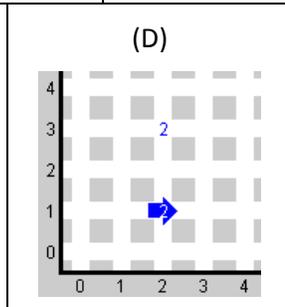
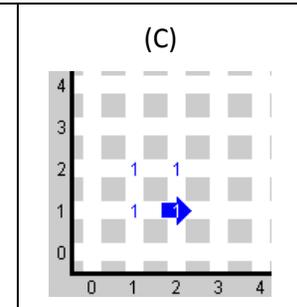
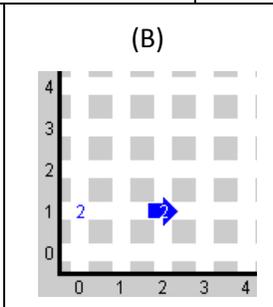
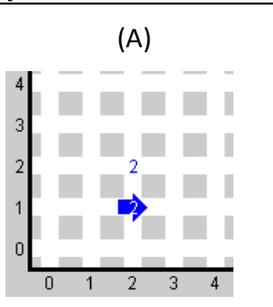
class program {
    void dropbeeper(n) {
        iterate (n) {
            turnleft();
            move();
        }
        putbeeper();
    }
    program() {
        while
        (anyBeepersInBeeperBag()) {
            dropbeeper(2);
        }
        turnoff();
    }
}
        
```

Answer

```

class program {
    void dropbeeper(n) {
        iterate (n) {
            turnleft();
            move();
            putbeeper();
        }
    }
    program() {
        while
        (anyBeepersInBeeperBag()) {
            dropbeeper(2);
        }
        turnoff();
    }
}
        
```

Answer



LOOK UP THE NEXT CONCEPTS IN YOUR BOOK AND DEFINE THEM BRIEFLY (3 PTS).

RECURSION

PRED

SUCC

MARK WITH AN "X" ALL TRUE STATEMENTS. (36 PTS)

- A robot executes a pickBeeper instruction only when the beeper-bag is not empty.
- A turnoff instruction is executed at the end of each program.
- While and If control statements are used for repetition.
- All the three sections we've used so far have Open and Save command buttons.
- Before running a program you must compile it.
- Considering the conditions of the World, turning left may force Karel to shut off.
- Having a beeper in front of the robot, performing a move command will generate an error.
- If all instructions of the code are correct, there's no need to compile.
- Parameter is the number specified between parentheses when using *iterate* to create functions.
- If Karel has a wall in front of it, performing a move command will generate an error.
- In order to turn right, the instruction *turnleft ()* must be used three times.
- Java is the programming language used in this course.
- Java language is set by default.
- Karel application program allows both: step by step execution and continuous execution.
- Karel can perform two or more instructions simultaneously.
- Karel executes a pickBeeper instruction only when it is on the same corner.
- Karel is represented by a blue arrow.
- Succ and Pred are functions used in recursion.
- Karel's application program version installed in the classroom permits the identification of the programming language because Java instructions are in English and Pascal instructions are in Spanish.
- Logical errors are detected by the application.
- More than one beeper can be picked up or dropped at the same time.
- Pickbeeper will be executed only if the beepers-bag is empty.
- Programming commands are case sensitive, that is, misuse of upper case and lower case letters will generate an error during execution.
- Recursion is used when a function invokes itself in one of the statements its code.
- Programs and worlds created in Karel are autosaved.
- Recursion occurs when a module calls itself.
- Saved world files must be opened in the world section.
- Streets run horizontally and avenues run vertically.
- The *Compile* command button is found in the program and execution sections.
- The cursor is a green square located somewhere in the world.
- The cursor is used to locate beepers.
- The instruction *Putbeeper()* may be used if Karel has beepers in its bag.
- Functions and modules are mainly used to solve a complex problem by means of small tasks.
- The *iterate* statement may use parameters.
- The *Iterate* command may be used only with the *turnleft ()* instruction.
- The robot executes a move instruction only when the path is clear to the next corner immediately in front of it.
- The World section has a Save button but doesn't have a *Save as* command button.
- To execute a program is the same as to run the program.
- Two or more conditions can be tested simultaneously.
- Walls can be inserted with code.
- You may use copy and paste in the Program section.
- If /else is called a compound conditional statement because two actions can be specified under one single condition.