CSE 1061 **Introduction to Computing**
Lecture 4 + HW#2

Week 2

Fall 2015

Department of Computing
The School of EE & Computing
Adama Science & Technology University

OUTLINE

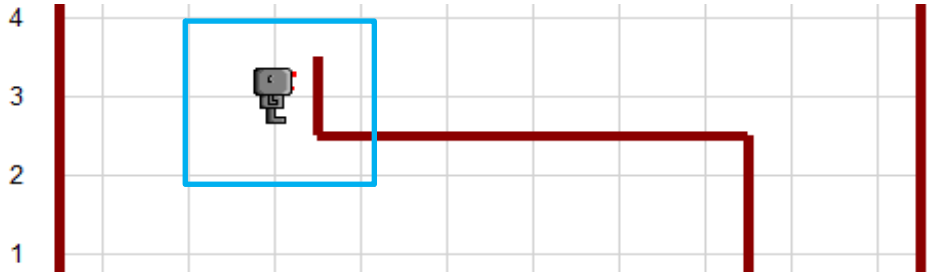
Conditionals

While-loops

Practice

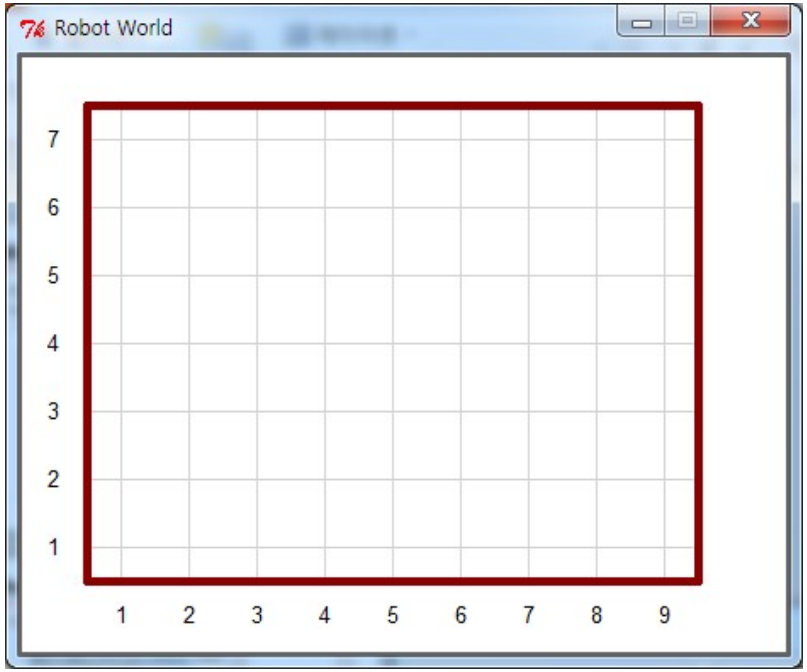
To check if there is a wall on the left, right, or front.

```
hubo.front_is_clear()   False
hubo.left_is_clear()     True
hubo.right_is_clear()   True
```



To create a grid

```
create_world(avenues = 9, streets = 7)
```



PRACTICE

ASTU

Read sections 10~13 in the robot note to do the following tasks:

- Harvest4 (page 7)*
- Plant1 (page 9)
- Hurdles3 (page 11)*

conditionals

- Hurdles4 (page 14)
- Harvest5 (page 15)*

while-loops

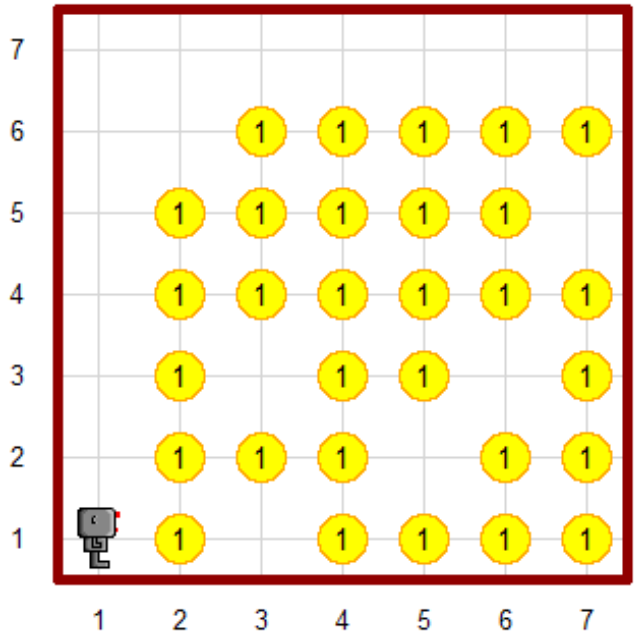
- ZigZag2 (page 17)

PRACTICE USING CONDITIONALS

ASTU

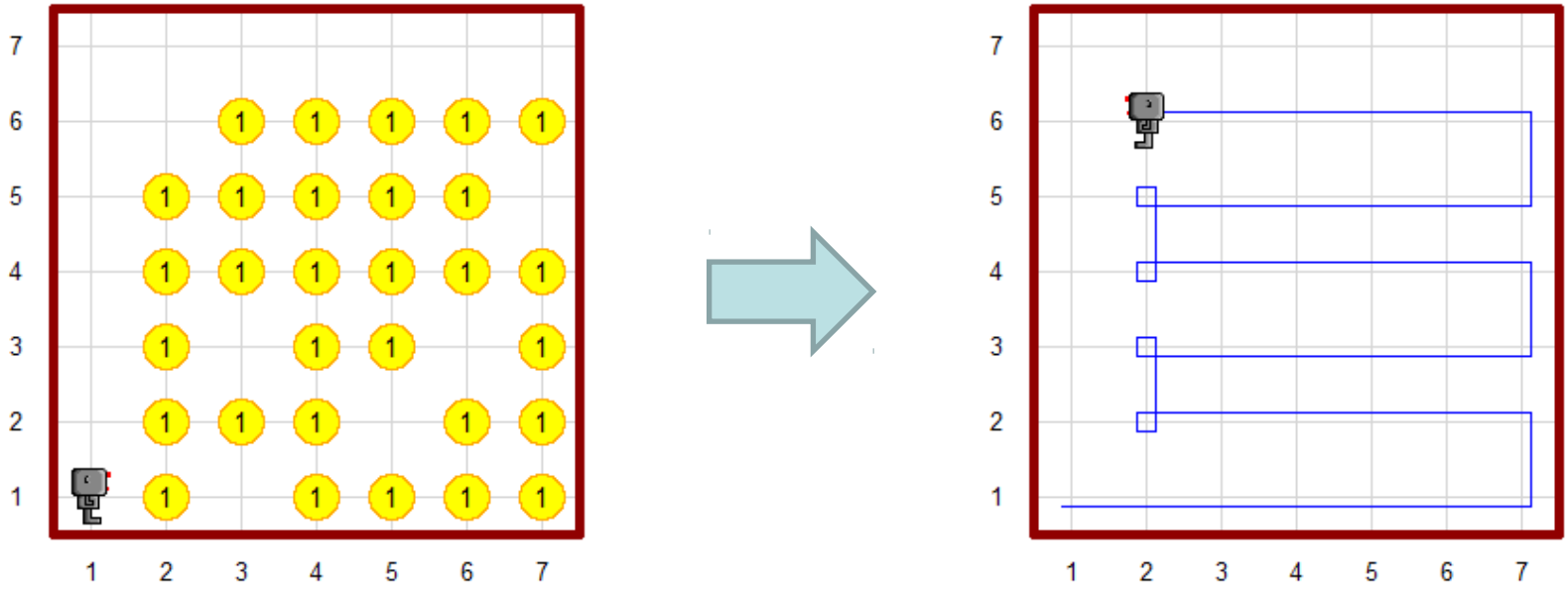
PROBLEM 8: HARVEST4*

Modify your program for the Harvest2 task so that the program works for the world harvest3.wld. Note that the new program should also automatically work for the original harvest.wld world.



Load the file, harvest3:

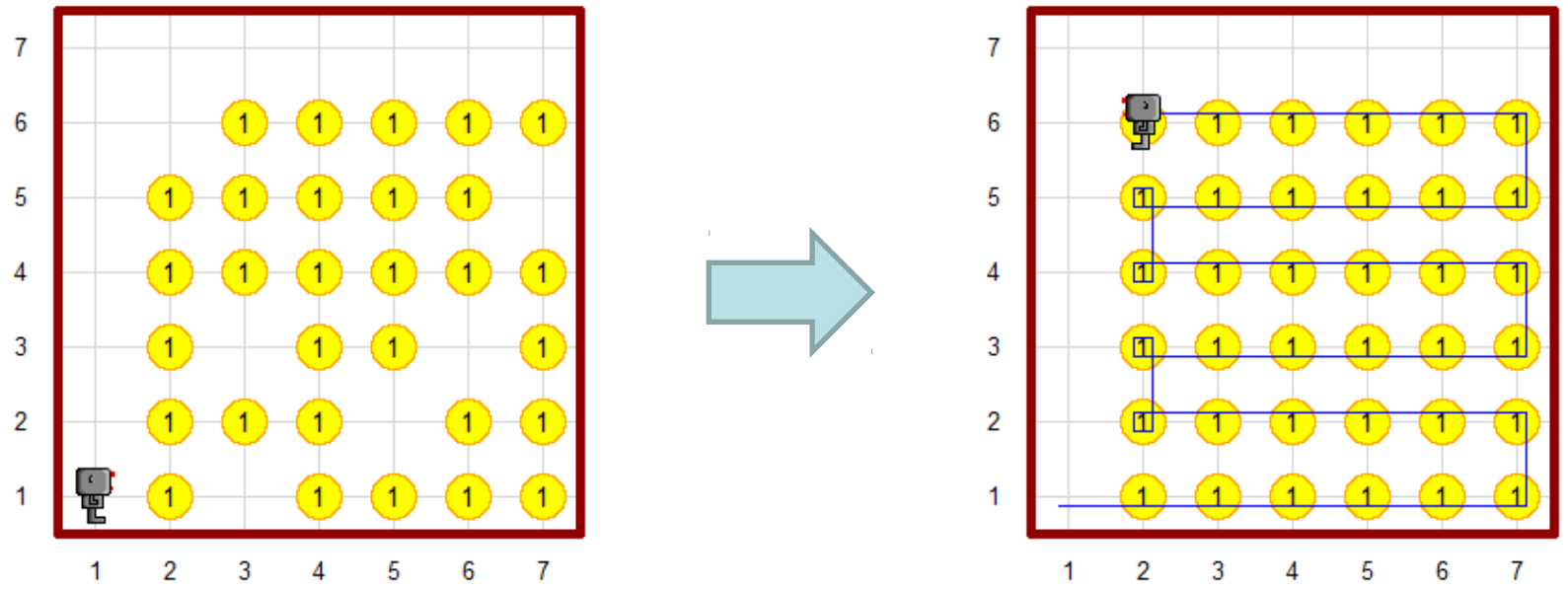
```
load_world("./worlds/harvest3.wld")
```


7
6
5
4
3
2
1
1 2 3 4 5 6 7

PROBLEM 9: PLANT1

Write a program that will plant potatoes so that the field will look like harvest1.wld at the end. Hubo should skip any spot where there already is a potato. (Note that you have to create your robot with sufficiently many beepers by using hubo = Robot(beepers=36).

Try your program with an empty world and with the worlds harvest1.wld and harvest3.wld.


7
6
5
4
3
2
1
1 2 3 4 5 6 7

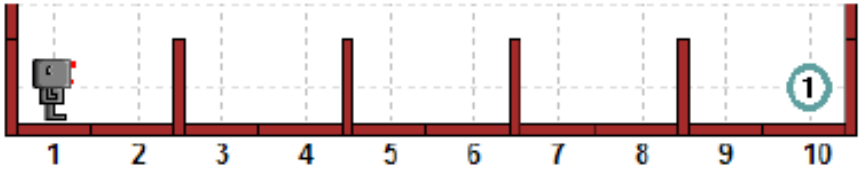
PROBLEM 10: HURDLES3*

Our robot has become quite good at jumping hurdles. Hubo now enters races of different lengths: short sprints and long races. He knows that he has reached the finish line when he gets to the position with a beeper. Assume that there are no races longer than 20 units.

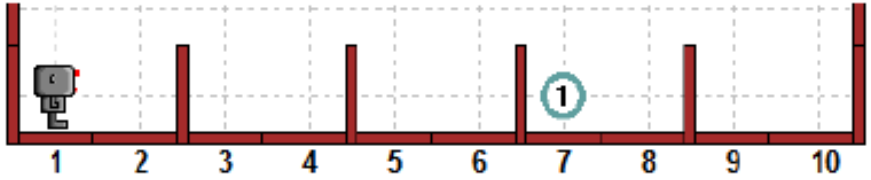
```
for i in range(20):
  move_jump_or_finish()
```

Use exit(1) in the sys module to finish a race.

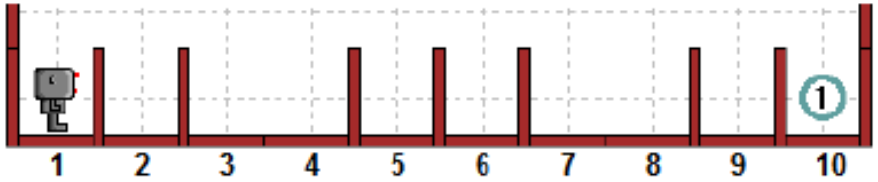
Below, you will find three such race courses; the world files are hurdles1.wld, hurdles2.wld, and hurdles3.wld :



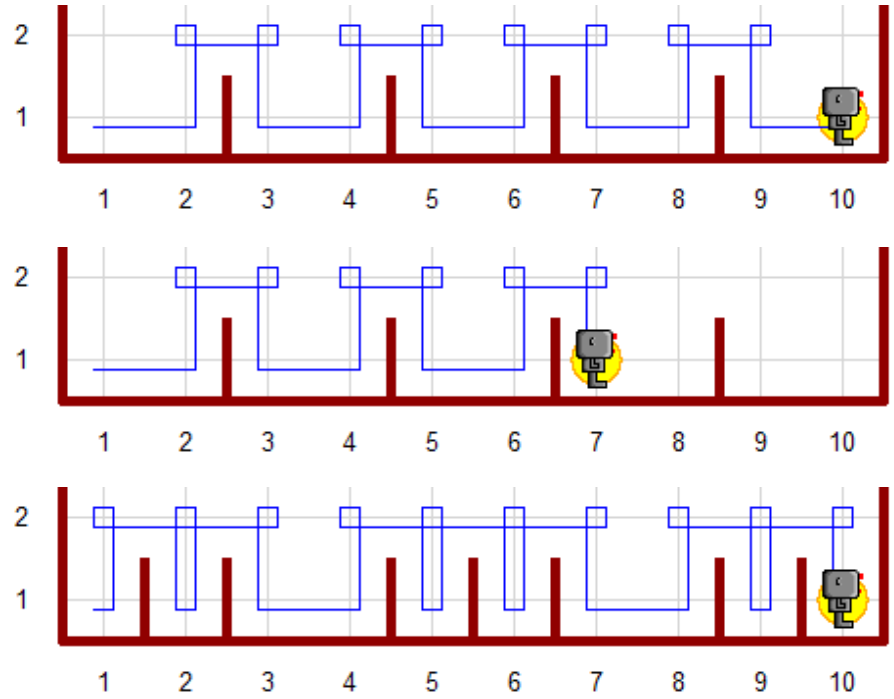
hurdles1.wld



hurdles2.wld



hurdles3.wld



hurdles1.wld

hurdles2.wld

hurdles3.wld

How to write a function jump_one_hurdle() for jumping over a hurdle ?

USING WHILE-LOOPS

ASTU

Homework 2 for Week Lecture 2: (problem 11 & 12)

Write a program which does

the following problem 11 and 12:

due date: 1 week from next Lab.

Submit to: ykchung99@gmail.com

Submition method:

- your source program-file (with id & name)

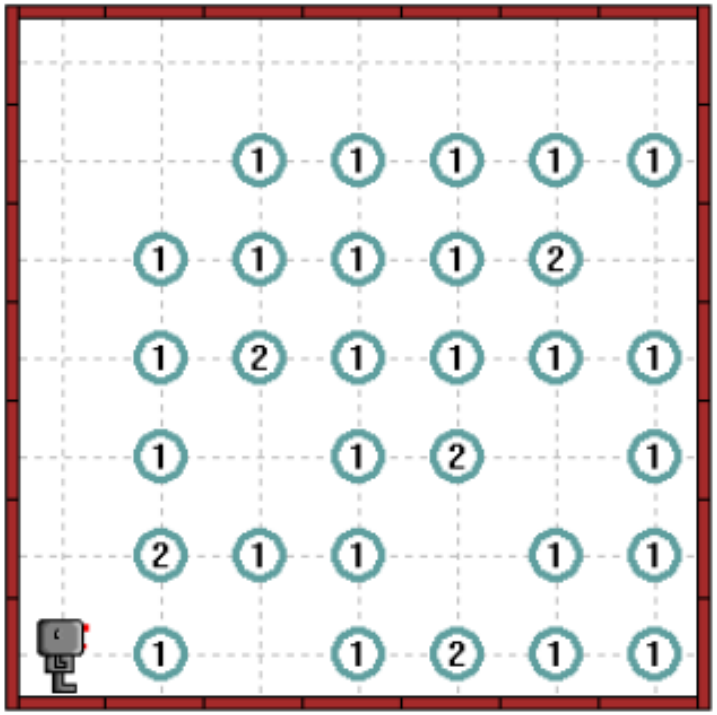
- e-mail or Lab-assistant using USB memory

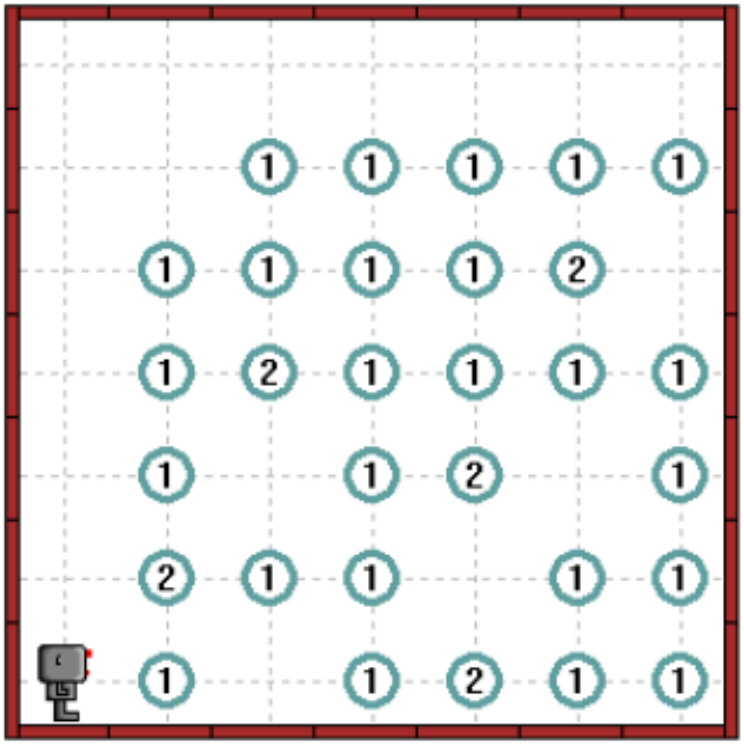
HW2-1) PROBLEM 11: HURDLES4

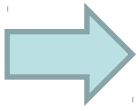
Use a while loop with “hubo.on_beeper()” to rewrite the hurdles3 program so that you don't have to use a for-loop of fixed length.

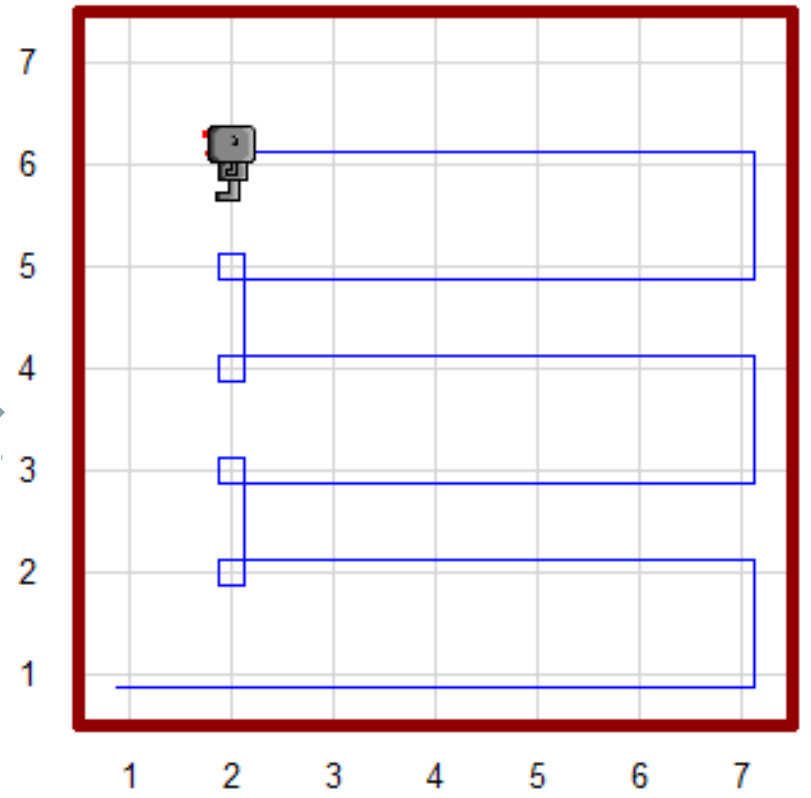
HW2-2) PROBLEM 12: HARVEST5*

Modify your program for the Harvest4 task so that it also works when there is more thar one carrot on one place, as in world file harvest4.wld below. All carrots must be harvested, and it should also work for the previous worlds harvest1.wld and harvest3.wld.




1 1 1 1 1
1 1 1 1 2
1 2 1 1 1 1
1 1 2 1
2 1 1 1 1
1 1 2 1 1



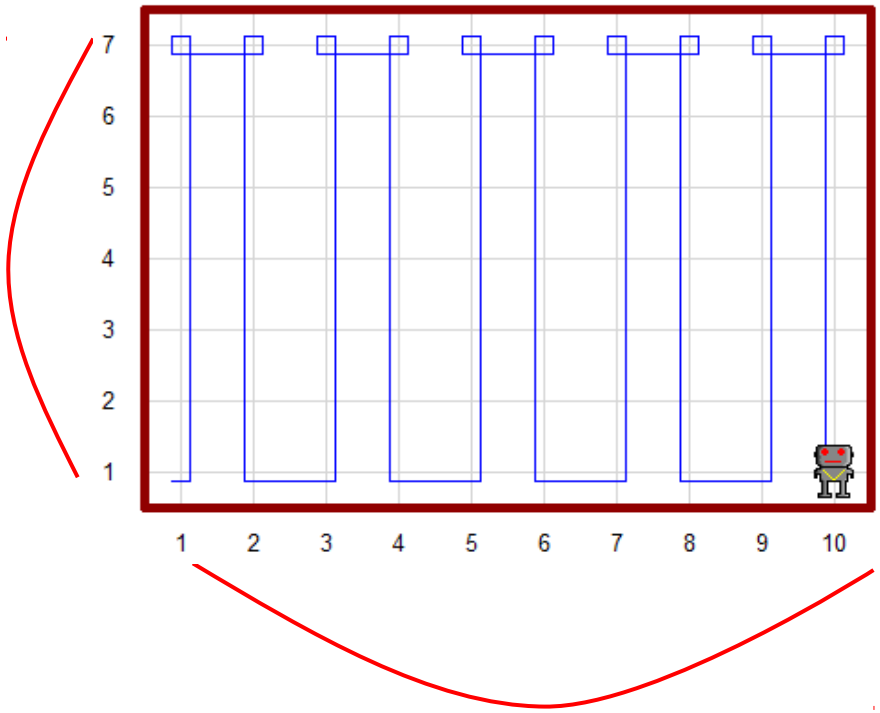

7
6
5
4
3
2
1
1 2 3 4 5 6 7

PROBEM 13: ZIGZAG2

Rewrite your program for ZigZag1 so that the robot can visit an empty world of any size in zigzag fashion. Make sure that the program works for even and odd numbers of streets and avenues. (You can assume that there are at least two streets and at least two avenues.)

Use randint() in module random to generate random numbers, e.g., random.randint(2, 20).

The number of avenues is even or odd. So is the number of streets.



Use while-loops.

Summary

Through **2D robot control** we learned:

- **conditionals**: `if, if~else, and if~elif~else`
- **iterations**
  - for-loops
  - while-loops
- **assignment**, e.g., `hubo = Robot()`
- **functions**

You have picked up the main constructs for programming.