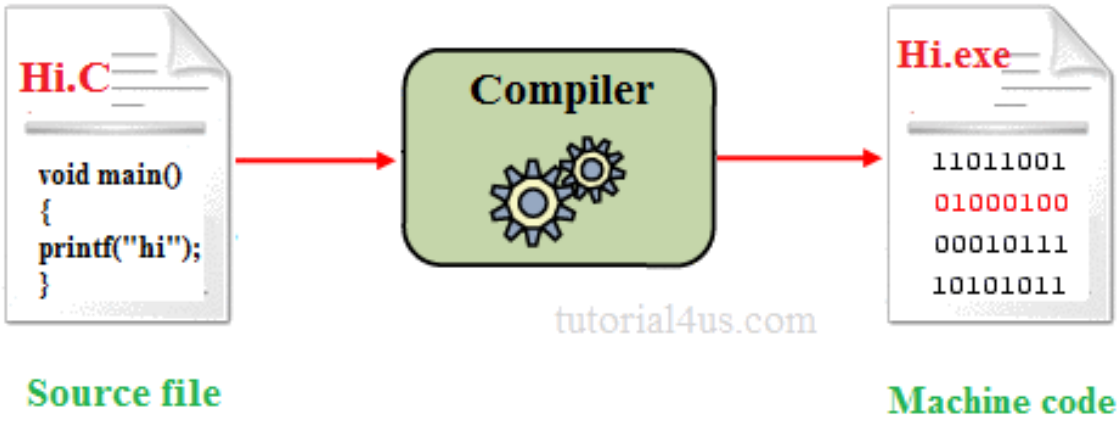
Vex Robotics Summer Program Day-2

Introduction to Programming

- What is programming?
  - Programming is writing vocabulary and a set of grammatical rules for instructing a computer or machine to perform specific tasks.

Introduction to Programming

- Where do we program?
  - Programmers program on a compiler. A compiler is a program that converts instructions into a machine-code or lower-level form so that they can be read and executed by a computer. It acts a translator for the computer.

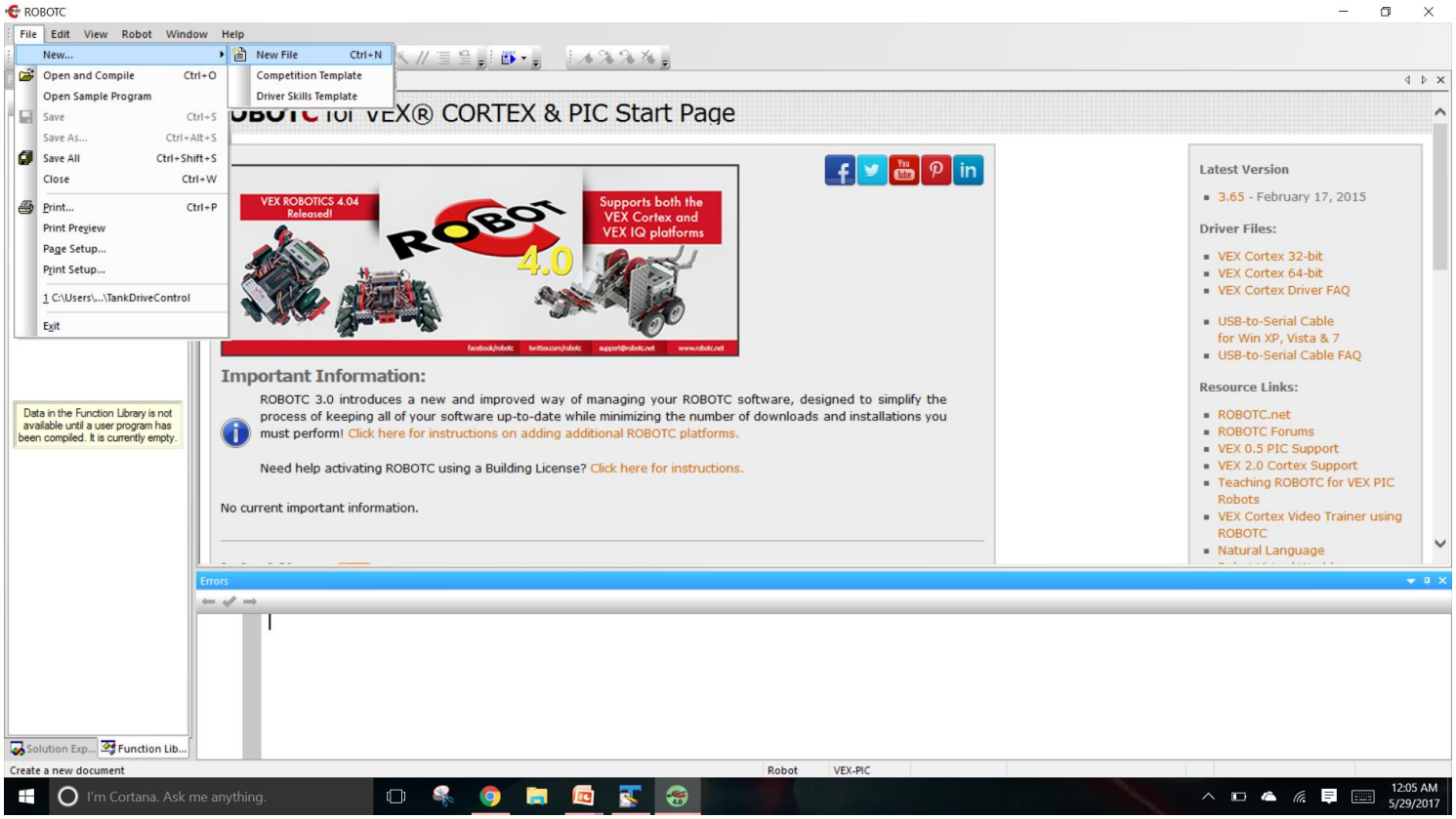


Introduction to Programming

- There are many programming languages but for Vex Robotics we will be focusing on C

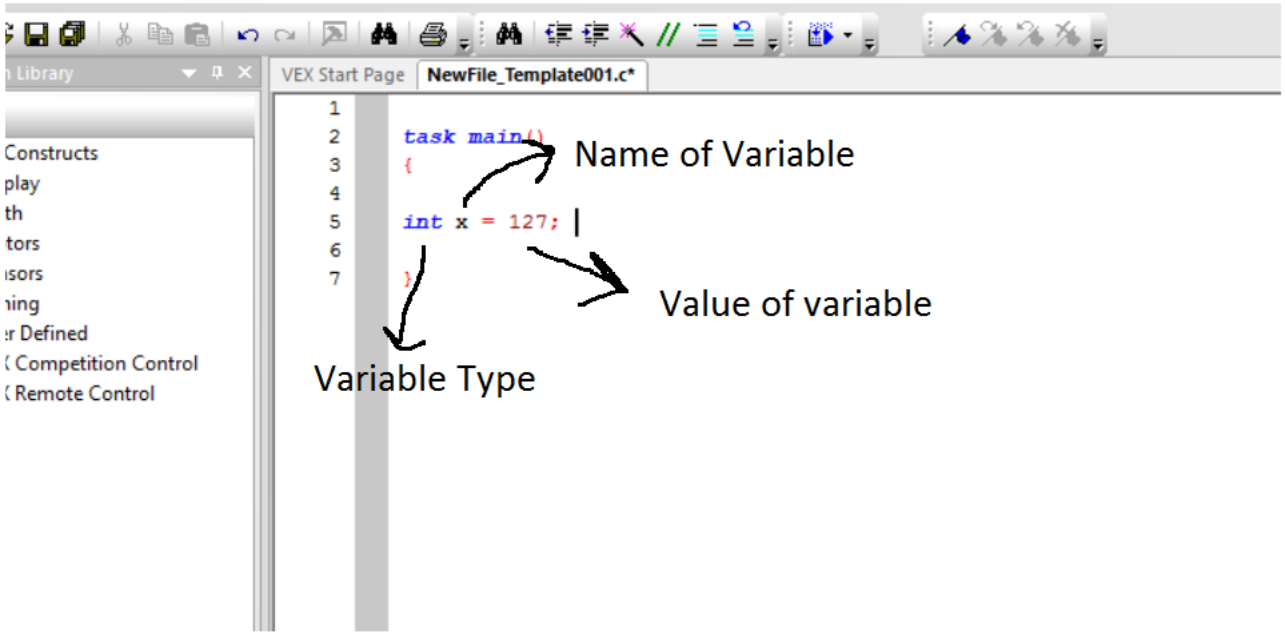
Basic Programming

- Open up Robot C and Create a new file for our first project



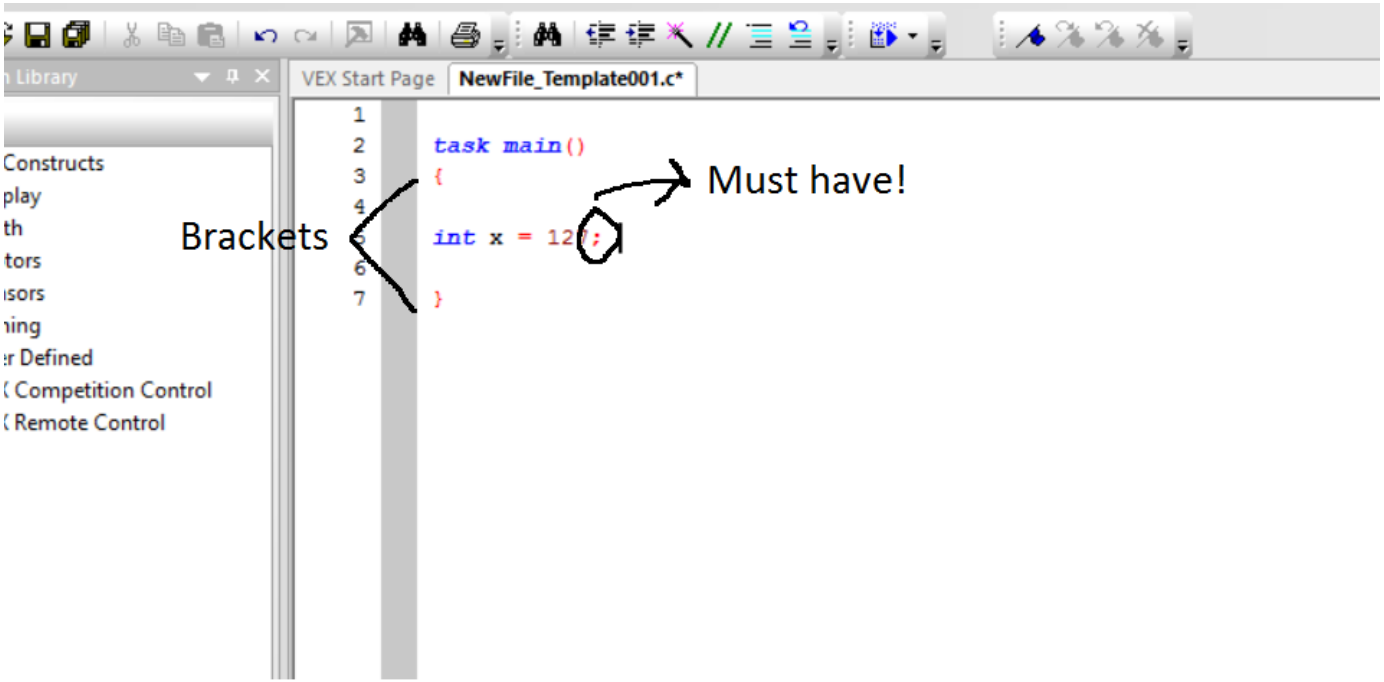
Basic Programming – Integers

- An int or an integer is a type of variable that stores a whole number between -32768 to 32768.



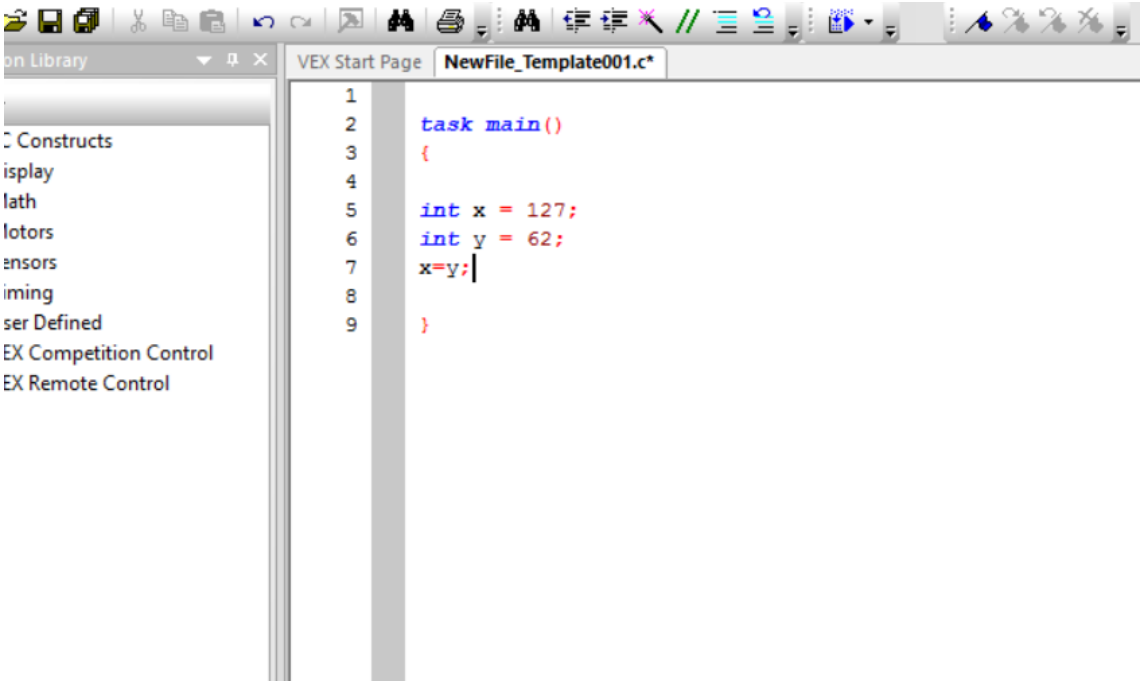
Basic Programming – Quick Tips

- Make sure code is in the bracket, compiler only reads what is in the main task
- Never forget semicolon at the end of any statement!!



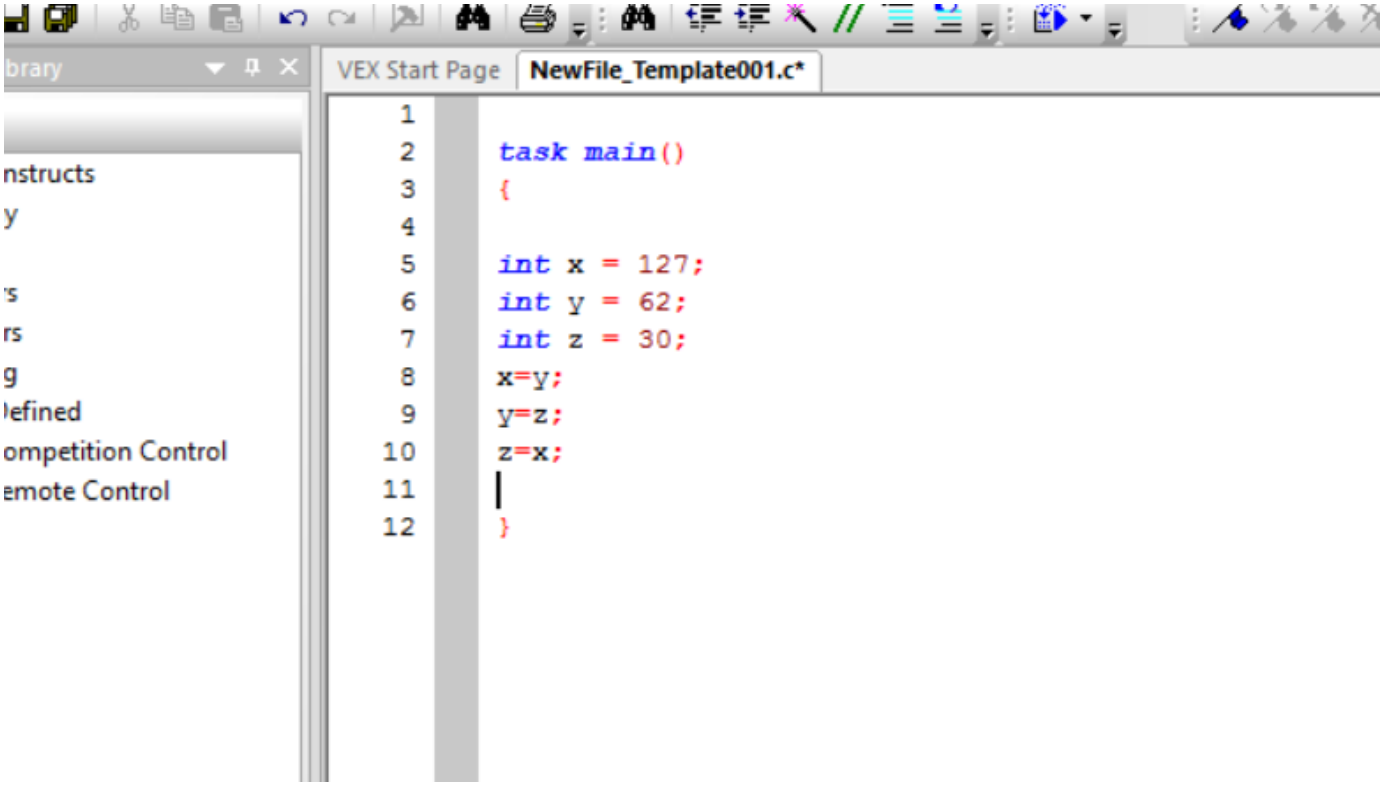
Basic Programming – Integers

- Whenever compiler reads an integer, the first integer is set equal to the value of the second
- After running the program below, both x an y will be equal to 62



Basic Programming – Practice

- After running this program, what is the value of x, y, and z?

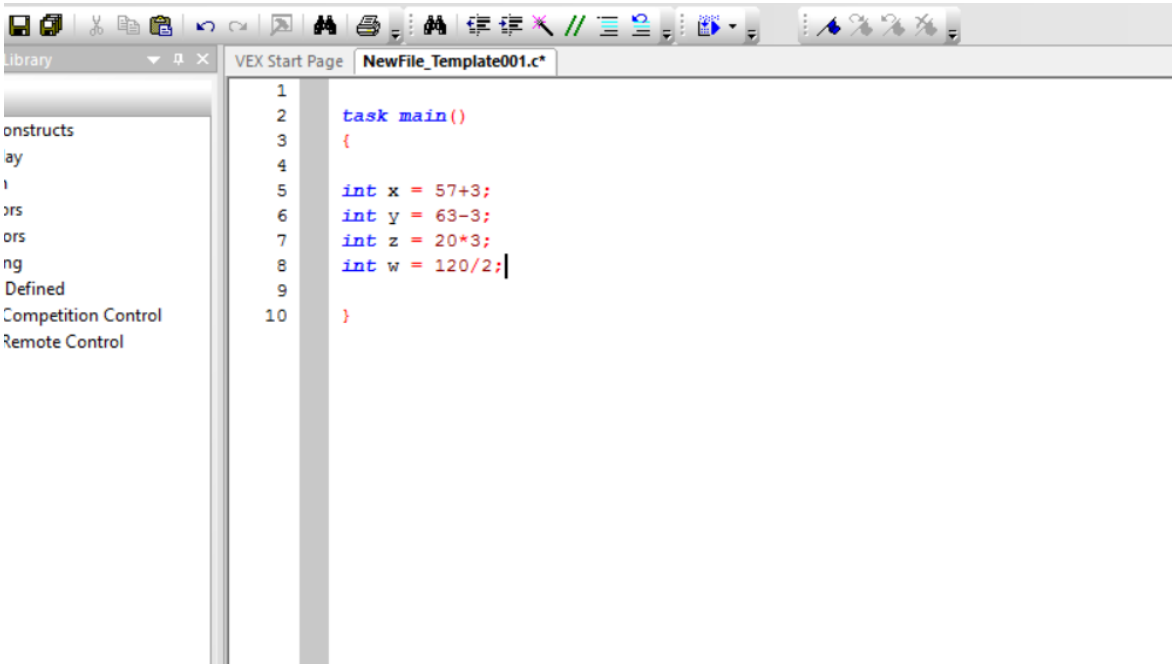


Basic Programming – Practice

- Answer
  - x = 62
  - y = 30
  - z = 62

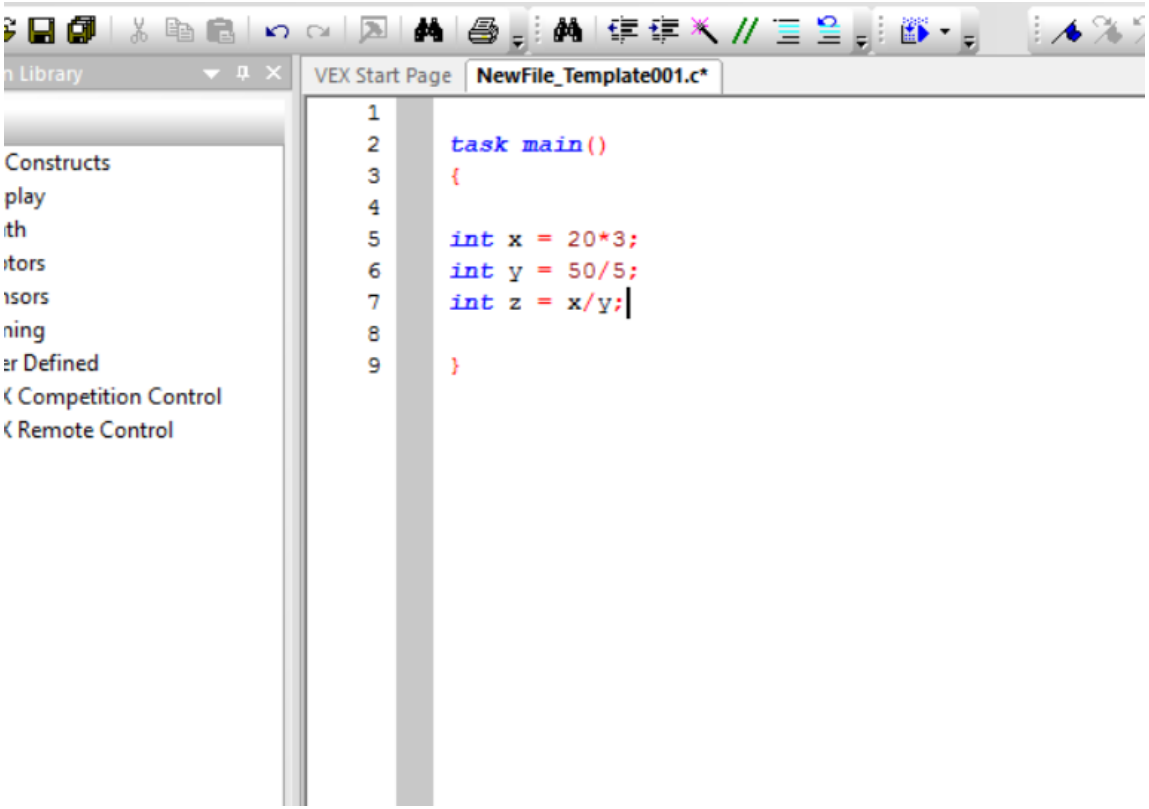
Basic Programming – Operations

- With integers, you can do basic mathematical operations such as addition, subtraction, multiplication, and division using “+”, “–”, “*”, and “/” operators
- In the scenario below, x, y, z, and w are all equal to 60



Basic Programming – Practice

- What does z equal to?

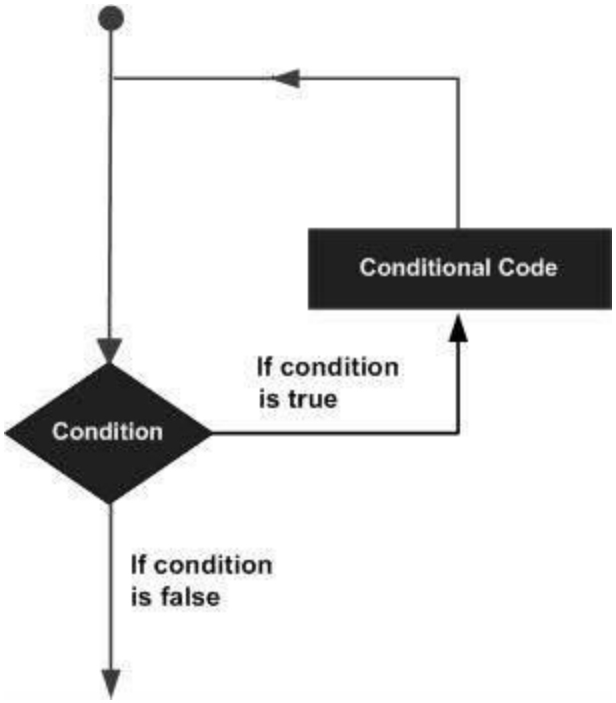


Basic Programming – practice

- Answer:
  - z = 6

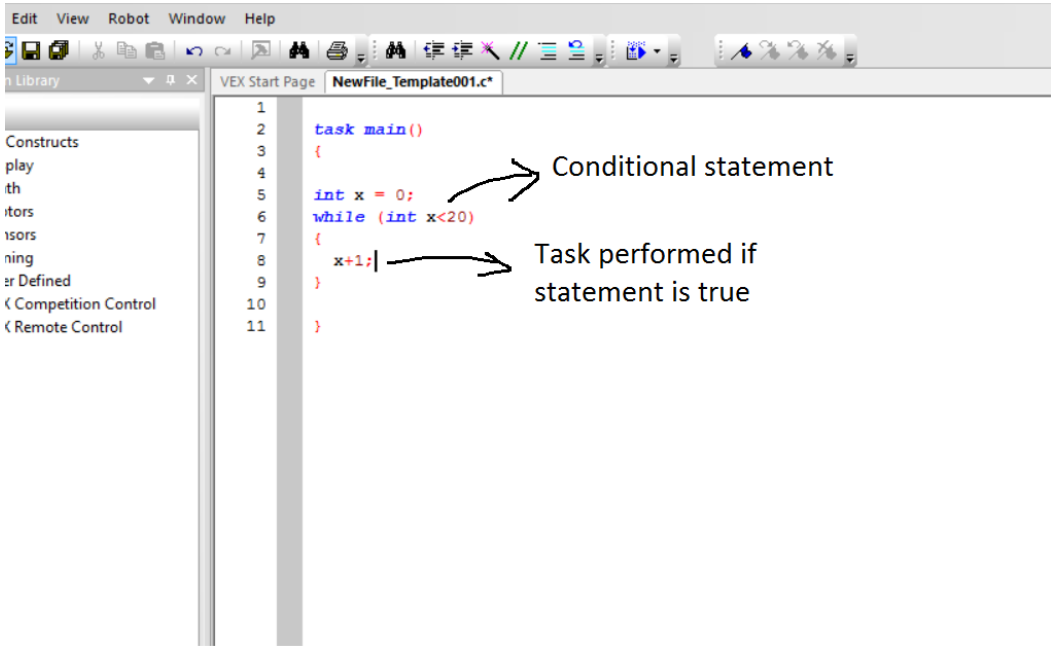
Basic Programming – Loops

- What is a loop?
  - A loop is a sequence of instruction s that is continually repeated until a certain condition is reached



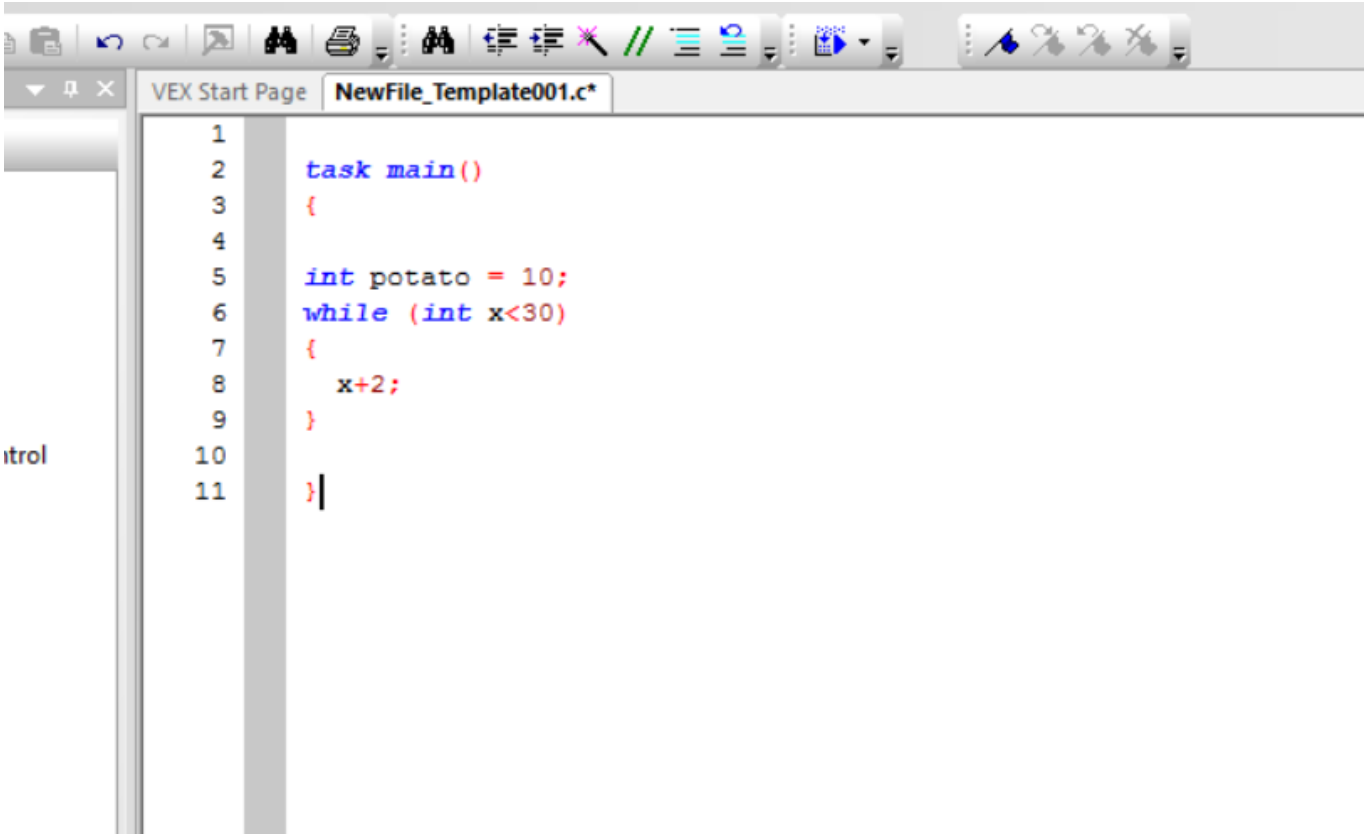
Basic Programming – While Loops

- In a while loop, the statement in parenthesis is the condition and what's in the brackets is the operation
- In the scenario below, x starts at zero, and the loop will continuously add 1 to x until it reaches 20. Thus the statement loops 20 times



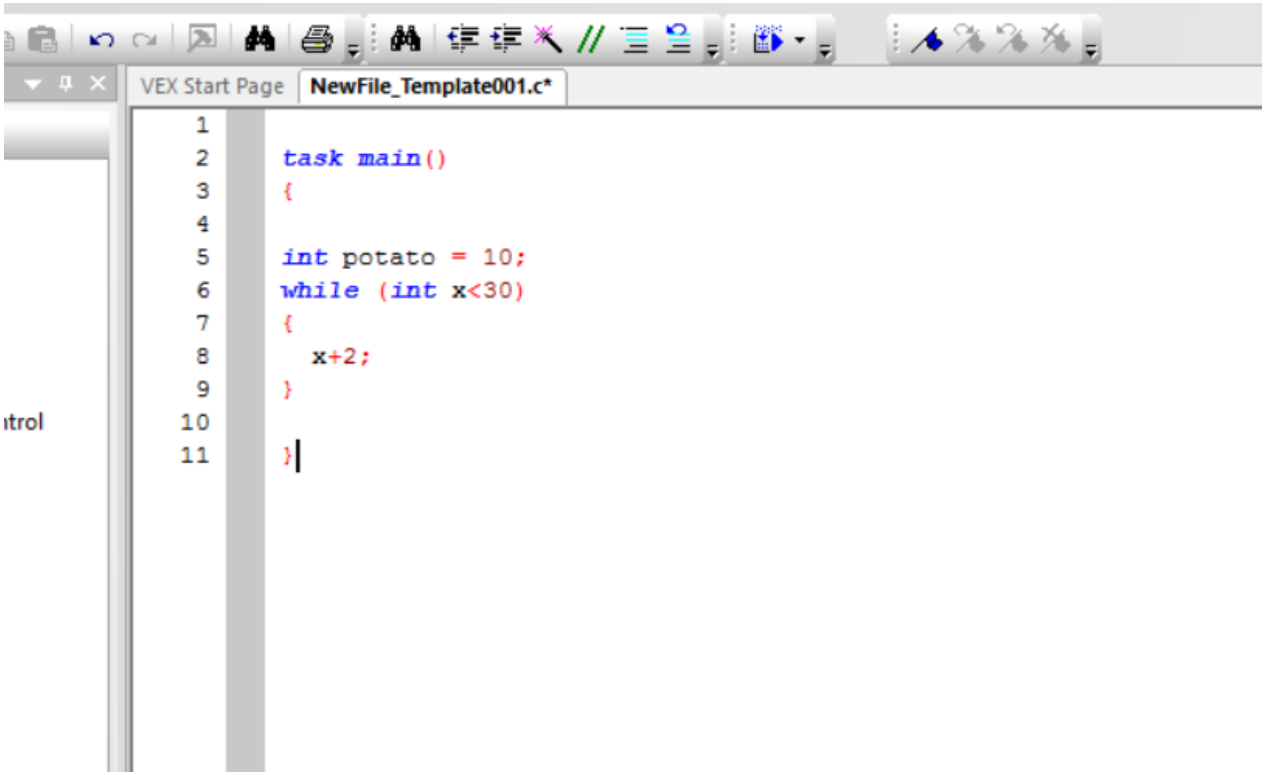
Basic Programming – Practice

- How many times does the loop run below?



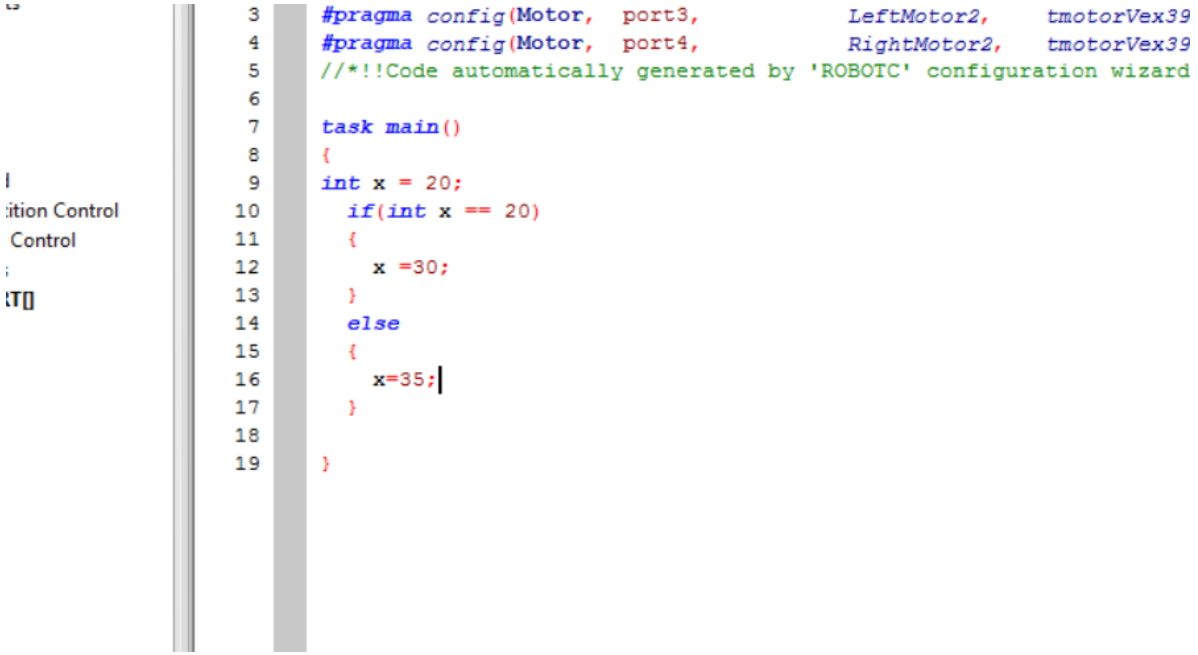
Basic Programming – Practice

- Answer:
  - 10 times



Basic Programming – if/else statements

- The if/else statement executes a block of code if a specified condition is true. If the condition is false, the else block of code can be executed
- In conditional statements, we use double equal signs instead of single
- In the code below, what is the value of x after running the code?



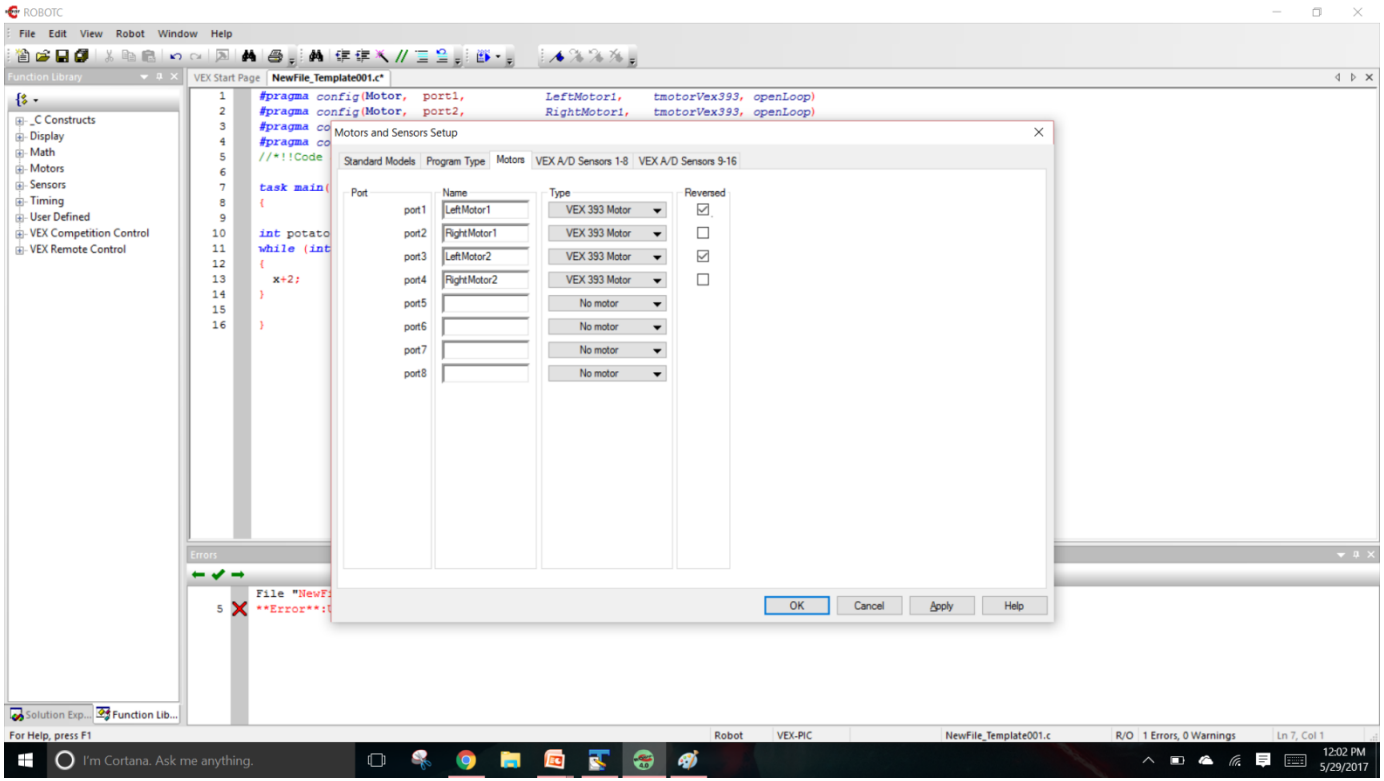

```
3  #pragma config(Motor,  port3,           LeftMotor2,    tmotorVex39
4  #pragma config(Motor,  port4,           RightMotor2,   tmotorVex39
5  //*!!Code automatically generated by 'ROBOTC' configuration wizard
6
7  task main()
8  {
9  int x = 20;
10   if(int x == 20)
11   {
12     x =30;
13   }
14   else
15   {
16     x=35;
17   }
18
19 }
```

Basic Programming – if/else statements

- Answer:
  - 30

Robot Programming – Motors

- First thing we will program are the motors
- To set it up, click “robot” → “motors and sensors setup”, and name four motors as named below. Set each of the motors to type “393 motor” and reverse two of them.

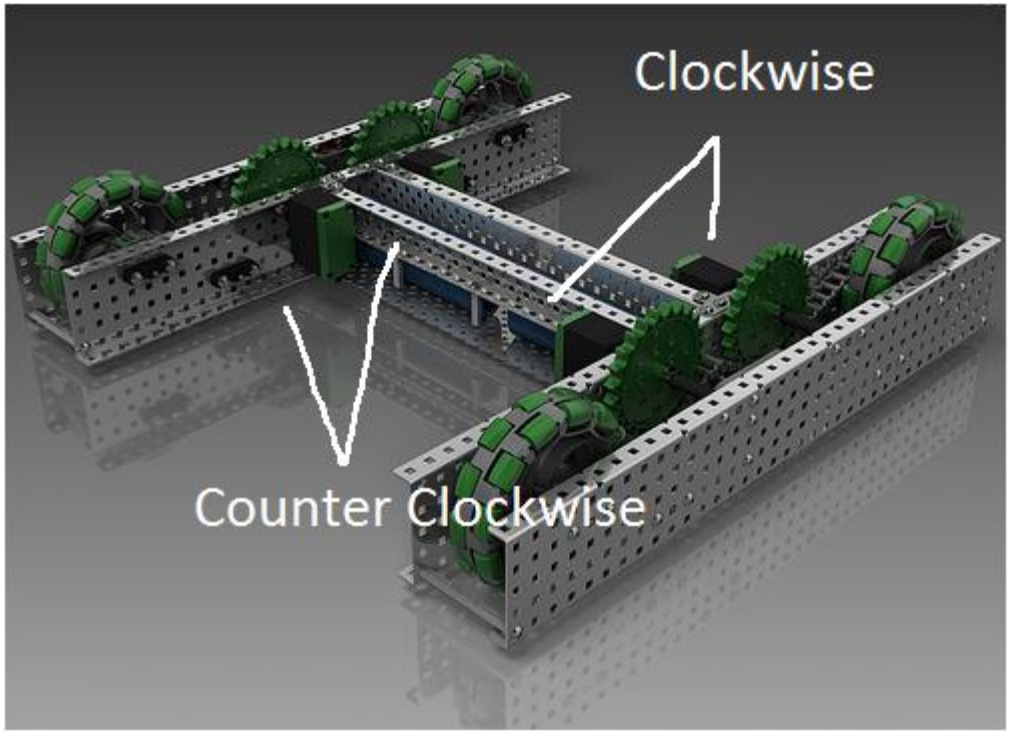


Robot Programming – Motors

- If we are programming the base, then why are two of the motors in reverse? How does the structure impact the direction of rotation?

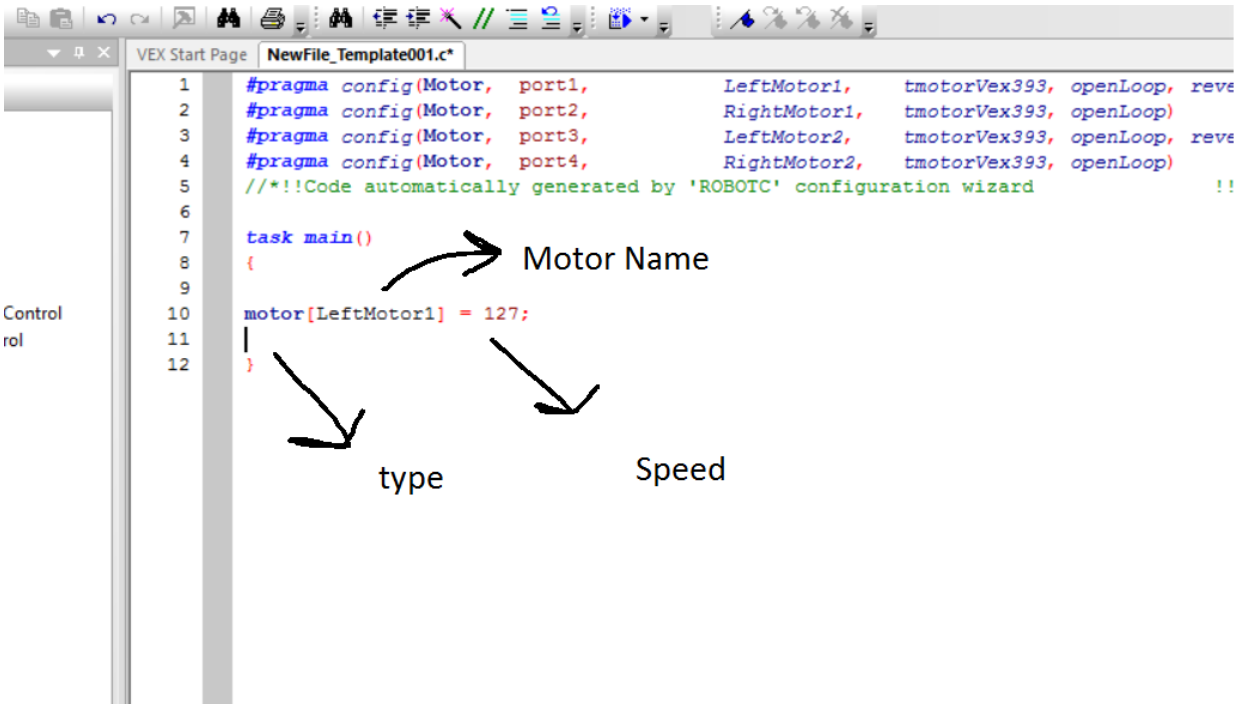
Robot Programming – Motors

- Answer
  - Because of the way the motors set up mechanically, they have to spin in opposite directions to move forward



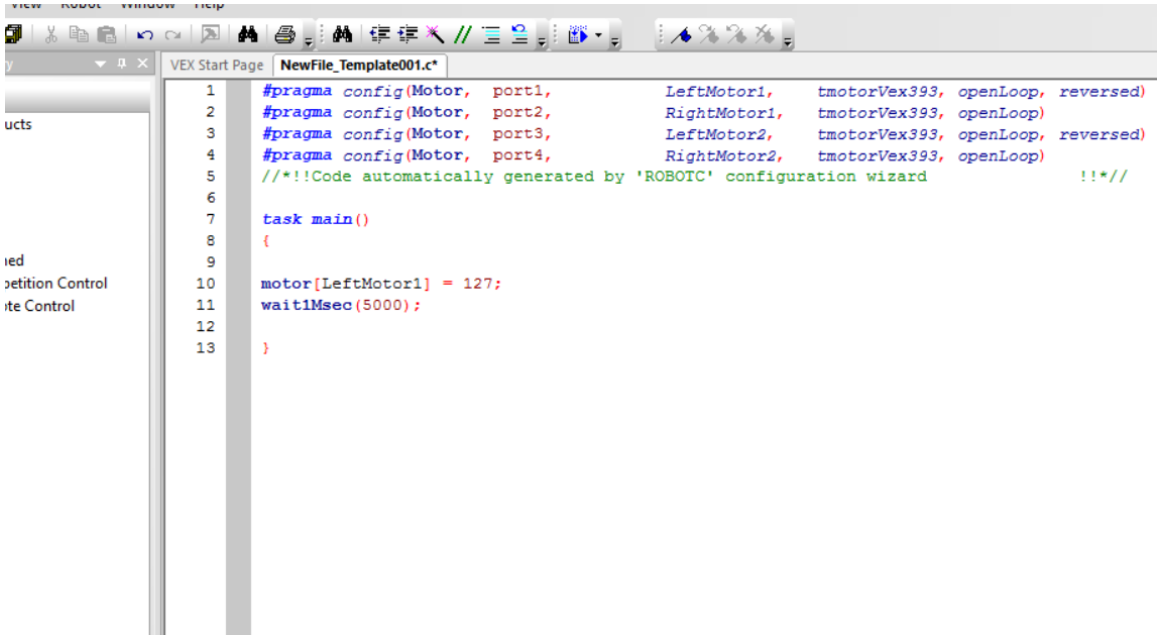
Robot Programming – Motors

- To program the motors, in the main task, we write "motor[*motorName*] = speed;". The value we put for the speed is any number between -127 and 127. 127 being the fastest clockwise and -127 being the fastest counter clockwise.



Robot Programming – wait1Msec

- wait1Msec
  - Method that tells motors how long to spin
  - Measure in milliseconds
  - The code on the bottom will move the motor for 5 seconds

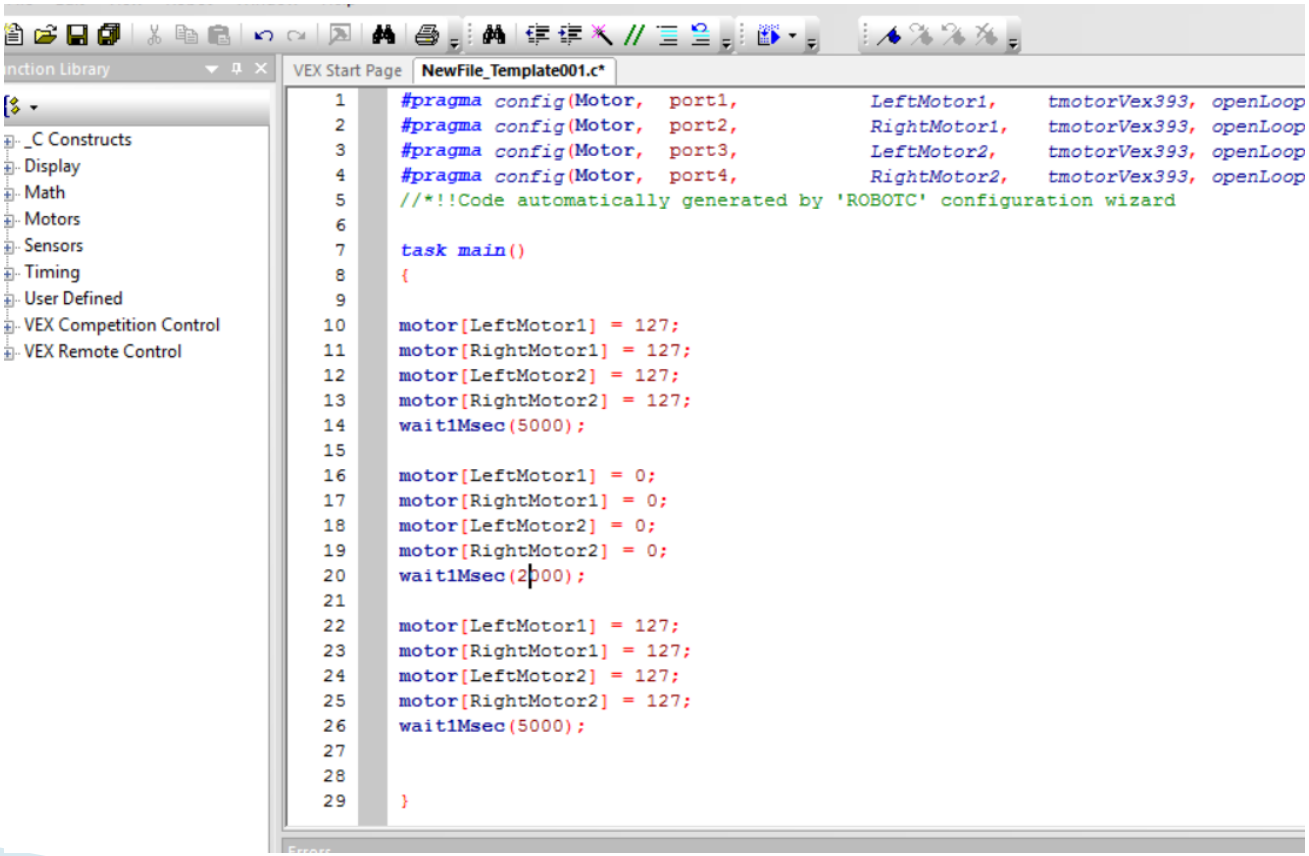


Programming – wait1Msec

- Program base to move for 5 seconds, stop for 2 seconds, and move for 5 seconds. Then upload the code and test on robot

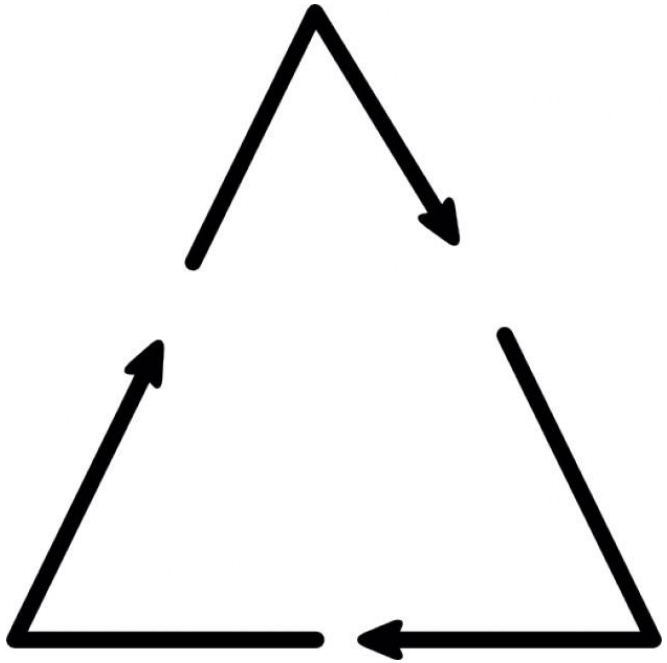
Robot Programming – wait1Msec

- Code should look like this



Robot Programming – Challenge

- Program the robot to move in a triangle

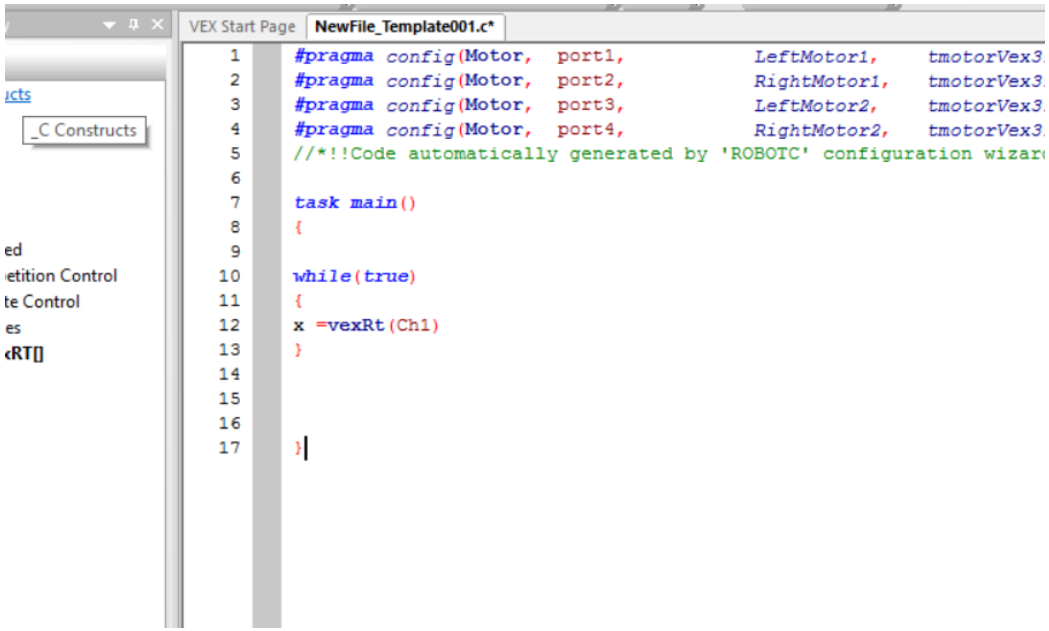


Robot Programming – Driver Control

- Now we get into how to program with remote control
- Everything must first be in a continuous while loop, we set while(true) to always run it

Robot Programming – Driver Control

- Method VexRT()
  - This method returns a value between -127 and 127 depending on how far the thumb sticks are pressed
  - In the bottom code, if thumb stick is pressed all the way up, x will equal 127, if it is pressed all the way down, x will equal -127
  - In the parenthesis, we write the channel number, which can range from ch1, ch2, ch3, ch4 depending on what the axis

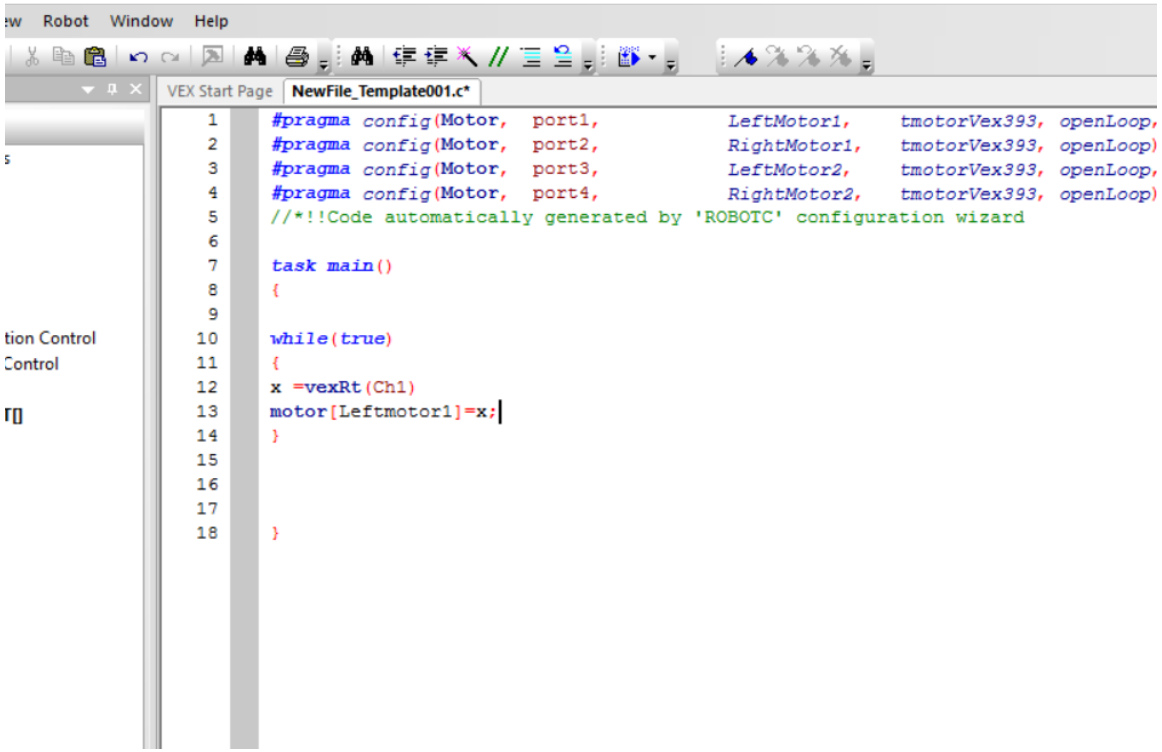


Robot Programming – Practice

How would you use the method VexRT and variables to move a motor when you the move thumb sticks on a remote?

Robot Programming – Practice

- Answer:
  - Setting the variable equal to VexRT and setting the speed of the motor equal to that variable

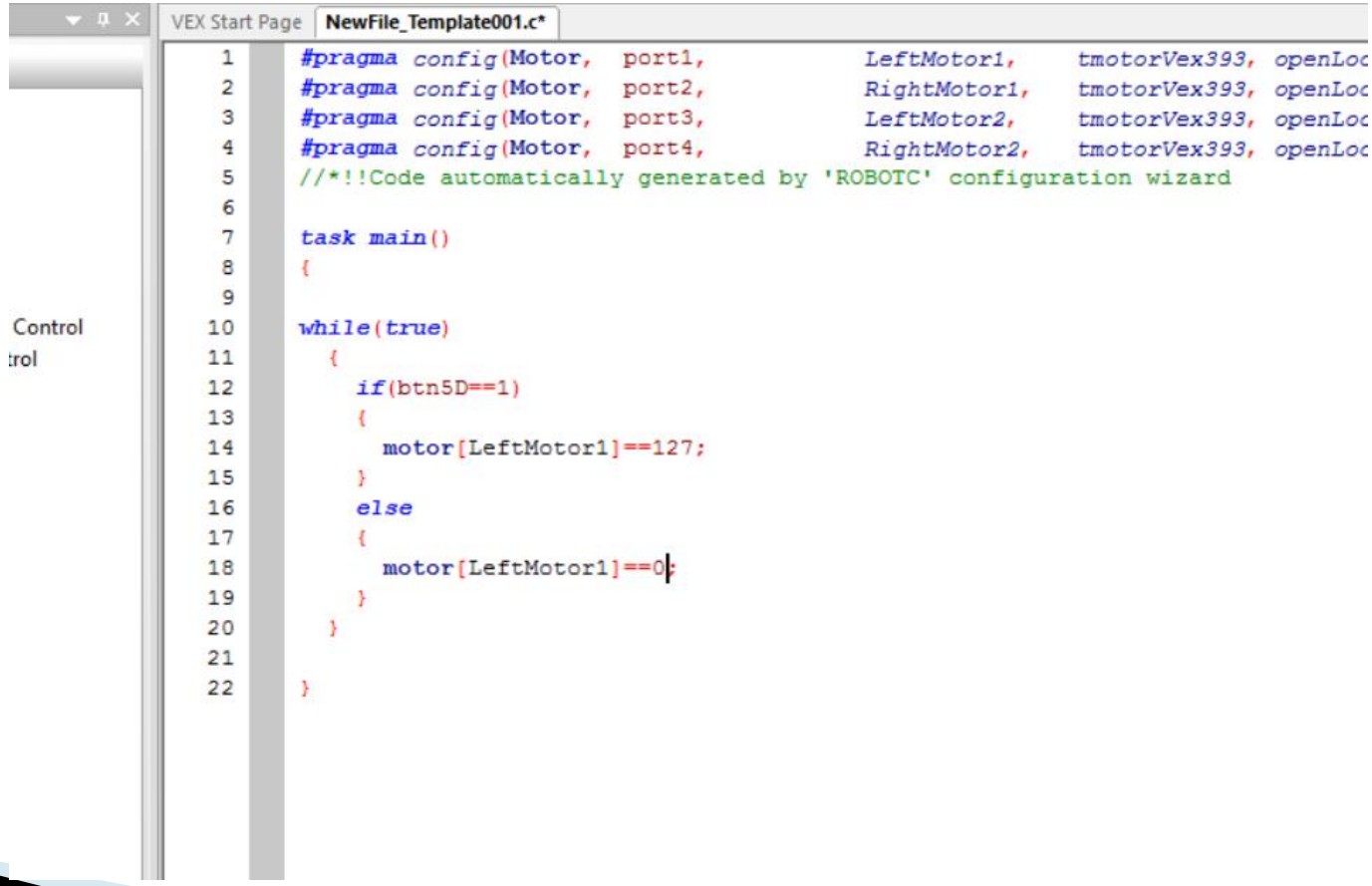


Robot Programming – Button

- Buttons are like thumb sticks except they take return value of 1 or 0
- We use Btn5U to indicate the uppermost button in group 5, and Btn7R for right most button in group 7 and so forth.
- Using if statements, program a motor to move forward if the lower button in group 5 is pressed

Robot Programming – Button

- Answer:



Robot Programming – Challenge

- Program the entire base to move forward, backwards, left, and right with remote control. Incorporate buttons and thumb sticks.