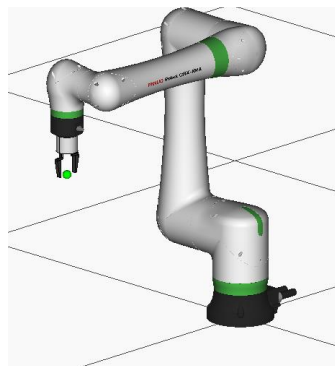


# Fanuc CRX 10IA Collaborative Robot: Curved Motion Type, For Loops, Labels, Jump Command and Call Command

By: Matthew Jourden  
Brighton High School  
Brighton, MI

1. Turn On
  - a. Controller
  - b. Tablet Teach Pendant (Tablet TP)
    - i. Verify Payload > Code = 1111 (NOTE: This code is setup by the school or company)
    - ii. Payload 1 > Yes > Ok > Ok
2. Open Program Home > Run Program to Place Robot in Default Position



3. Place Robot in Teach Mode (Top Right Corner > Open Rectangle

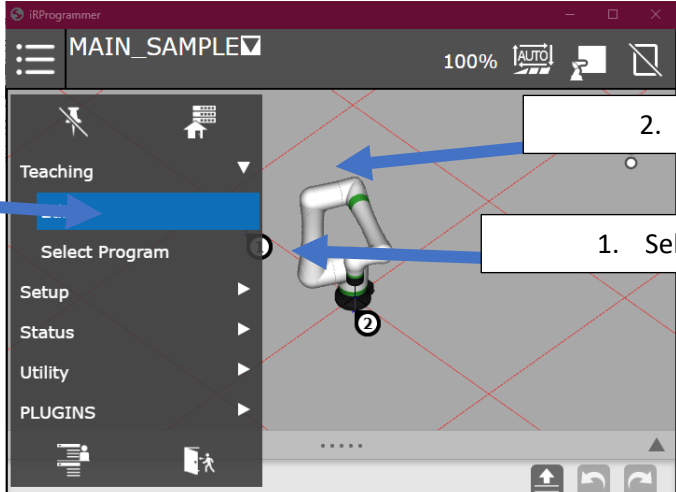


4. Reset any Faults

5. Select Menu Button

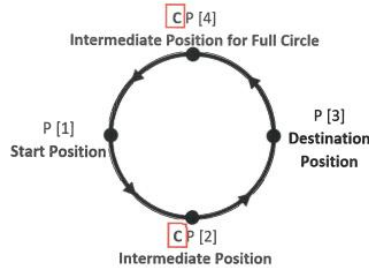


6. Select Teaching Down Arrow > Select: Select Program

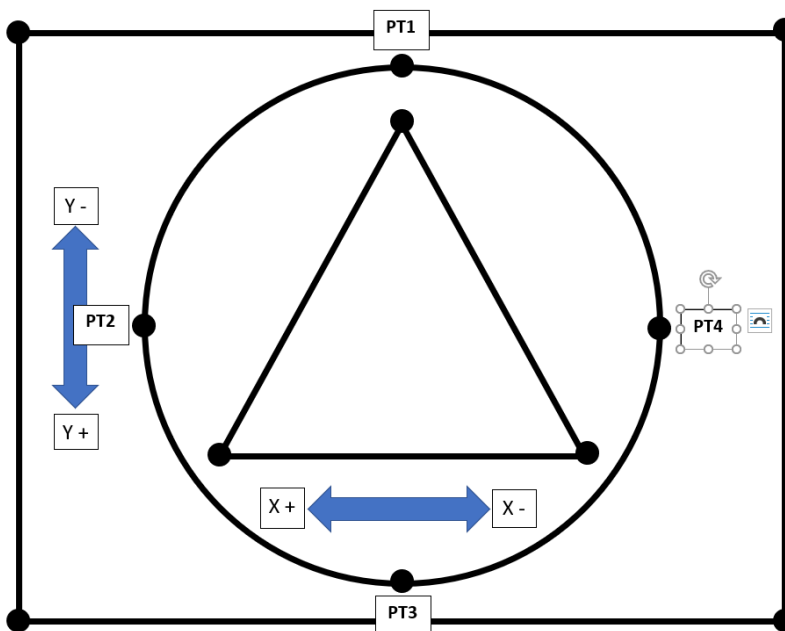


The screenshot shows the iRProgrammer software interface. The title bar reads "iRProgrammer" and "MAIN\_SAMPLE". The interface includes a menu on the left with options: Teaching, Select Program, Setup, Status, Utility, and PLUGINS. A blue arrow points from the "Teaching" menu item to a callout box that says "Opens Active Program". Another blue arrow points from the "Select Program" menu item to a callout box that says "1. Select: Select Program". A third blue arrow points from the "Teaching" menu's downward arrow to a callout box that says "2. Select Down". The background shows a 3D model of the robot arm in a virtual environment.

7. Select New Program > Program Name: Shape\_Circle\_”Class Hour”
8. Circular Motion defines an Arc. The Motion Type needs to have 3 Points to define an Arc
  1. Linear or Joint Leading to 1<sup>st</sup> Position
  2. Intermediate Position: Some Point along the Arc
  3. Destination Position: Final End Point of the Arc



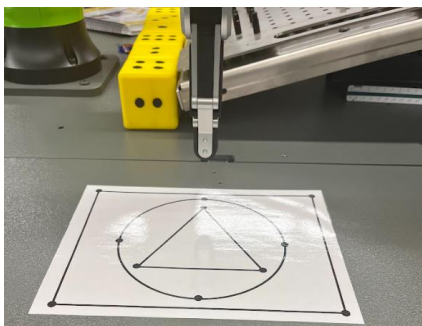
9. Circle Points with Coordinate Orientation



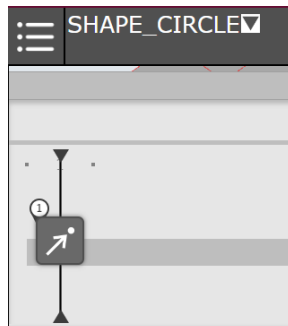
10. Movements

a. Point 1

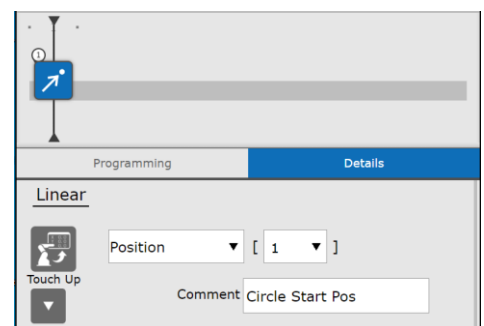
- i. Jog Robot to Point 1 on the Circle  
NOTE: Z Axis should be a minimum of 2 inches from the table
- ii. Drag a Linear or Joint Command on to the Timeline  
NOTE: Point 1 on the Circle becomes Position 1 in the program



Jog Robot to Point 1



Linear or Joint Command  
Drag Linear or Joint Command  
on to the Timeline



**Comments:** Adding Comments to  
the Position Point will make it  
easier to know where the robot is

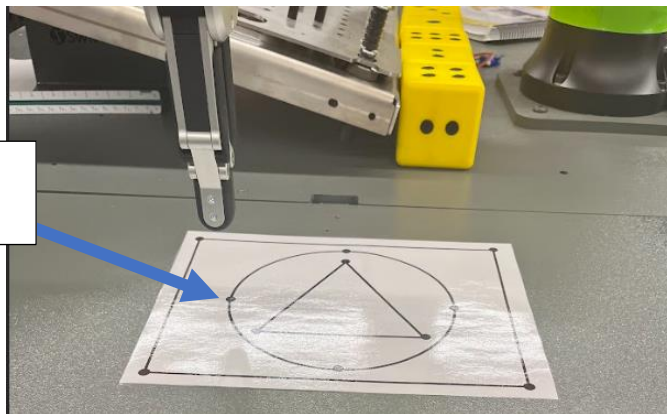
## b. 1<sup>st</sup> Half of the Circle

### i. Circle Point 2: Intermediate Position

1. Drag and Drop Circular Command on to the Timeline
2. Point 2 on the circle: This point is known as the Intermediate Position
3. Jog the Robot to Point 2 on the Circle

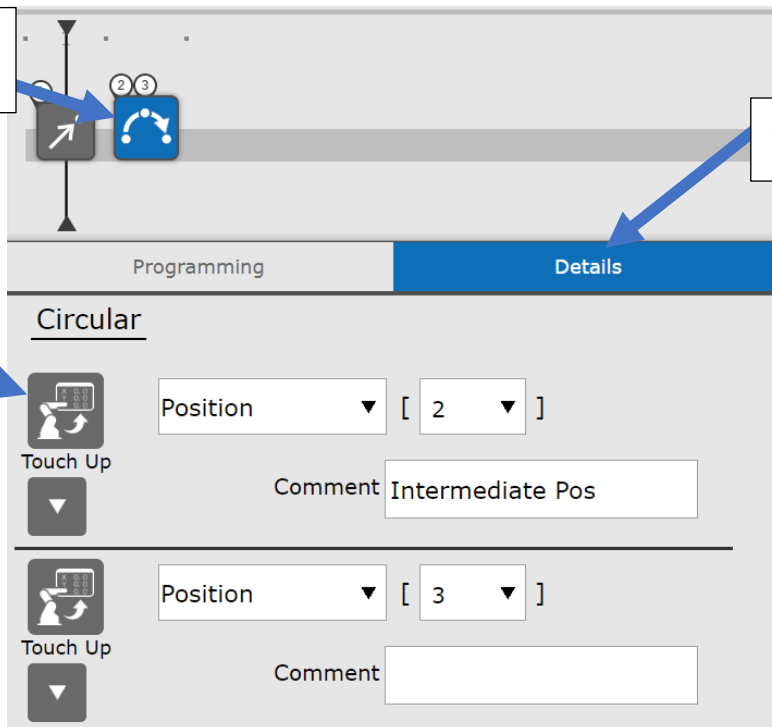


Jog Robot to Circle Point 2  
(Intermediate Position)



4. Select Circular Command on Timeline > Details Menu
  - Select Touchup next to Program Position 2

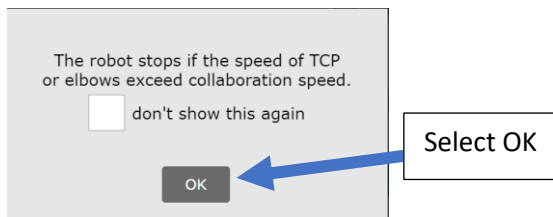
1. Select Circle Command  
on timeline



2. Select Details

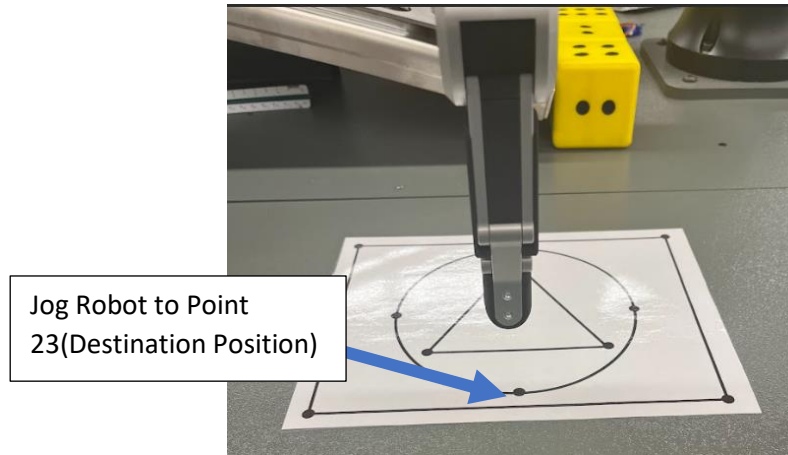
3. Select Touch Up for  
Program Position 2  
Intermediate Position  
(Circle Point 2)

- Error Message will Appear > Select OK  
Error Represents: If Robot Stops if the speed of TCP or Elbows Exceed Collaboration Speed: This message means if the robot is travelling at a high rate of speed and tries to decelerate and stop at a point then the Centrifugal Forces on the joints may exceed the 150N of force needed to fault the robot > Select OK (NOTE: This tutorial is running at a max of 100mm/s.)

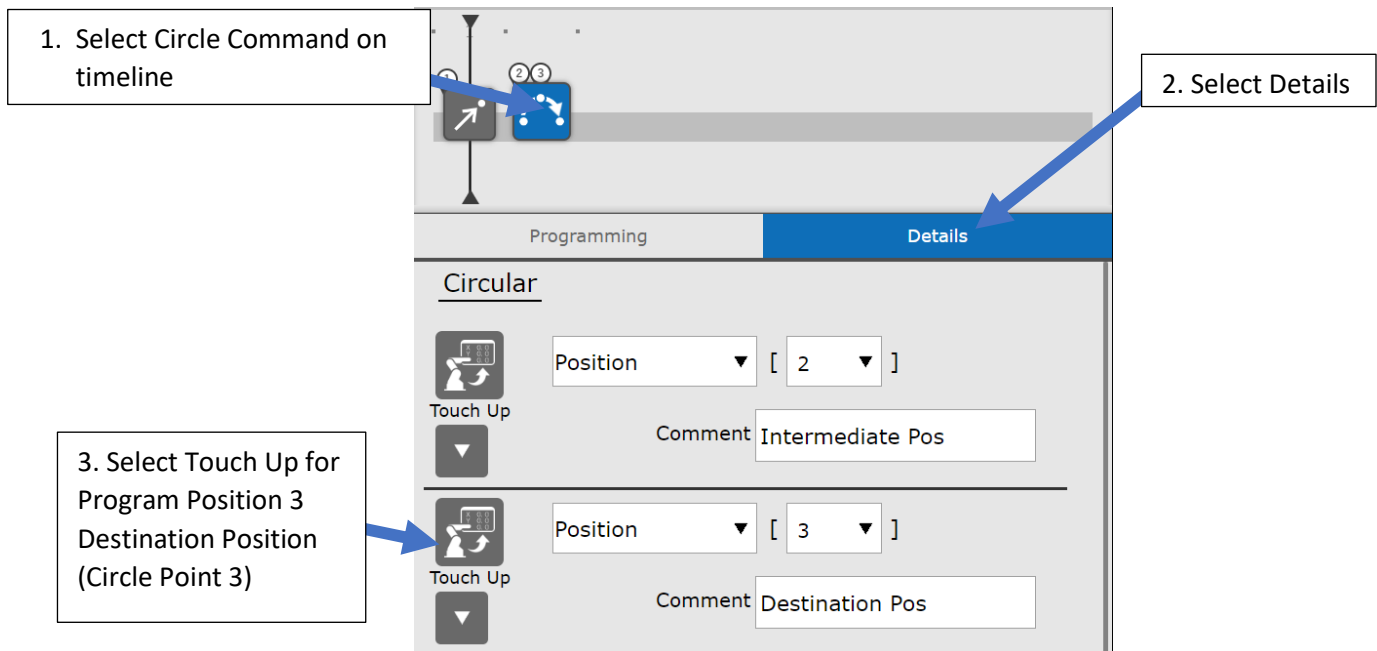


ii. **Circle Point 3: Destination Position**

1. Jog the Robot to Point 3 on the Circle (Destination Position)



2. Details Menu > Select Touch Up for Program Position 3



- c. Run the Program
- d. Optional: Turn the End Effector (Gripper) so the face arcs with the circle

11. Add the Second half of the Circle with another Circular Command

**NOTE:** Linear or Joint Command is not needed between the Circular Commands

- a. Repeat Steps B 1<sup>st</sup> Half Circle Points 3 and 4 of the Circle (Which become Program Positions 4 and 5 for the Program. NOTE: Position 5 for the circle is also Point 1/Program Position 1 on the Circle
- b. Run the Program  
NOTICE: How the Arm Stops at Point 2 (Program Position 3) of the circle before moving onto the 2<sup>nd</sup> half of the circle
- c. Find a Solution to make the circle move smoothly through all points without stopping at the Destination Points

# Submission: Shoot a Video of the Robot Running > Share Video with the Teacher

## 12. Looping

### a. Forever Loop: Program will need

Label Command: this will ID the Start Position for the Loop

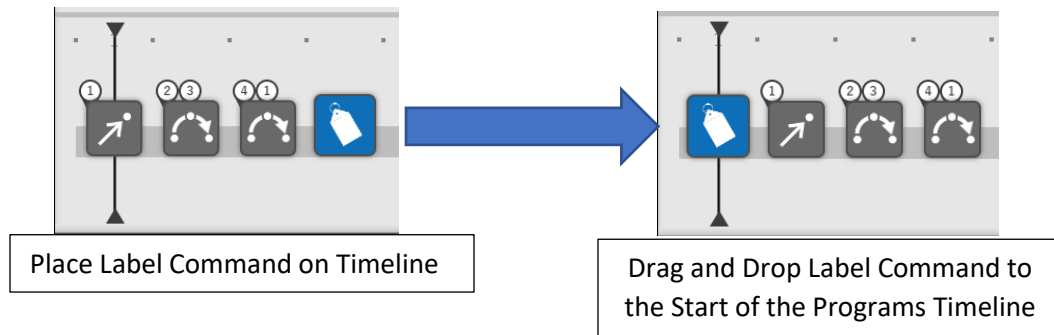
NOTE: Label can be placed anywhere in a program

Jump Command: Code Block that tells the program to move to a specific Label

#### i. Drag and Drop Label Command Block onto the Timeline

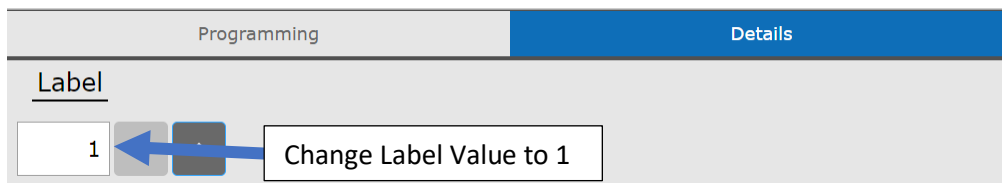


**NOTE:** Program may require the operator to drag at the end of the timeline > then drag to the front of the code

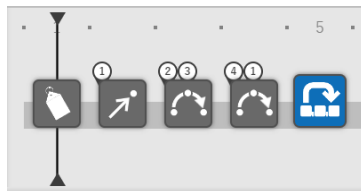


#### ii. Select Label Icon > Select Details > Change Label Value to 1

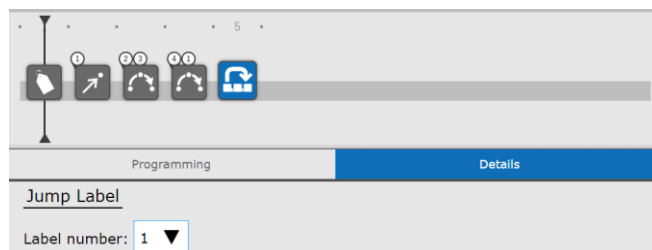
NOTE: Label Number starts at 0, Operator is required to change this value



#### iii. Drag and Drop Jump Command onto the End of the Timeline



#### iv. Select the Jump Command > Select Details Menu > Set Label Number to 1



#### v. Run the Program in Auto Mode

Notice the Program will run continuously until > Operator Presses the STOP Button.

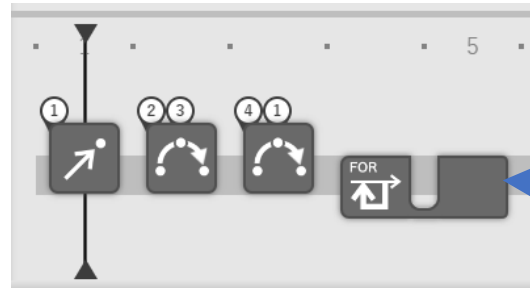
NOTE: Before running the program a 2<sup>nd</sup> Time > Double Tap the STOP Button to fully ABORT the Program and Reset the Timeline

- vi. After running the Program a few times > Delete the Label and Jump Command from the Timeline (Drag Code Block off the Screen to Delete)

## Submission: Shoot a Video of the Robot Running > Share Video with the Teacher

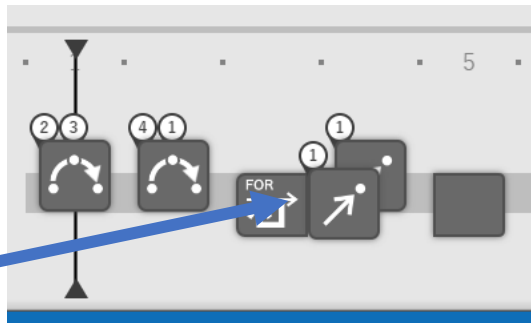
- b. **For Loop:** Allow the Operator to set a specific number of times to run the code

- i. Drag and Drop the For Command Block onto the Program Timeline



Place For Command on the Timeline

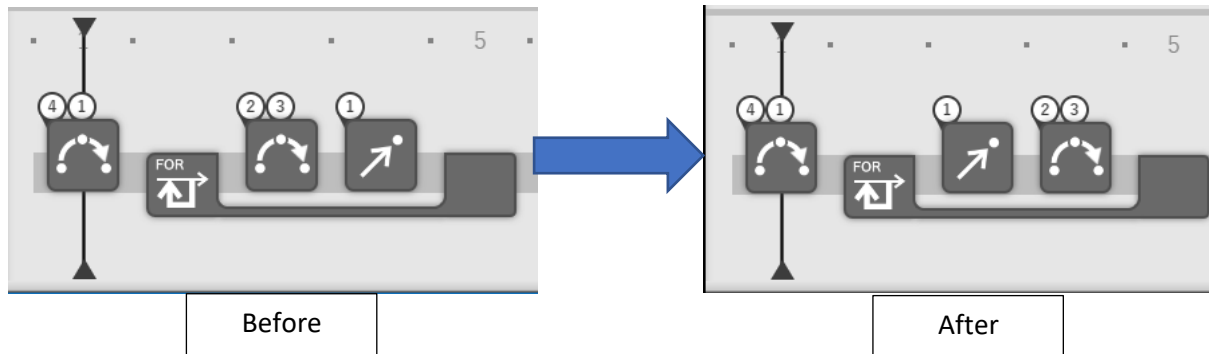
- ii. Code will need to be Dragged and Drop into the For Loop > When Dragging and Dropping Code Place the Cursor on or to the right of the FOR Icon



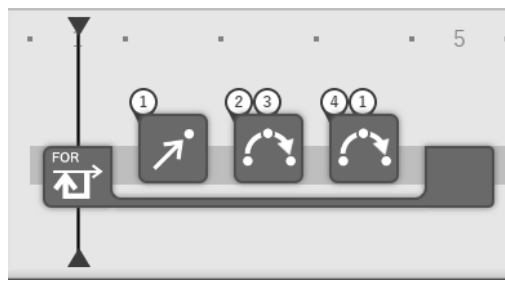
While Dragging Place Code Block On or to the Right of the FOR Icon

NOTE: Operator may need to organize the code to get the proper order. Code Blocks dragged into the For Loop will be placed at the beginning of the loop > Drag Code Blocks to desired order

- i.e. Place Linear Command 1<sup>st</sup> then Placed Circular Command 2<sup>nd</sup>, Finally Reorder Code



- iii. Final Code Order



iv. Select For Loop Command on the Timeline > Select Details

C++ For Loop Format: `for (int x = 1; x <=4; x++)`

Initial Value

Total Times thru the loop

Incremental Change each time thru the Loop

Loop Counter: Incremental Change each timeline thru the loop. (i.e. +1, +2 added to the initial value)

Initial Value: Starting value for the Loop

Target Value: End value to exit the loop (if counting by 1 then total time(s) thru the loop based on the Initial Value)

**Counter Type:** allows the Operator to Count up or Count down to a value



v. Adjust the values to the following

Loop counter: R [ 1 ]

Initial value: constant 1

Count type: To

Target value: constant 4

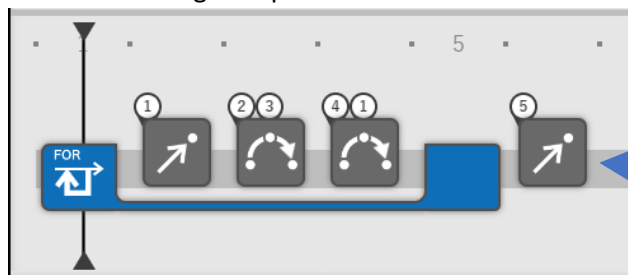
vi. Run the Program

**Submission:** Shoot a Video of the Robot Running > Share Video with the Teacher

### 13. Calling a Program

Calling a Program Allows the Operator to another program while operating in a current program. This allows the operator to not have to repeat movements that will be used multiple times in different programs without having to create the code blocks multiple times.

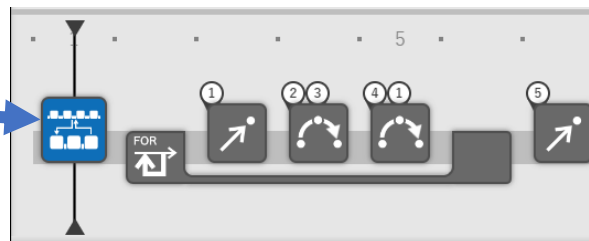
- a. Place a Linear or Joint Command that goes up in Z so the End Effector is above the Angled Table



Go up in Z to place the end effector (Gripper) above the Angled table

- b. Place the Call Code Block at the Beginning of the Program

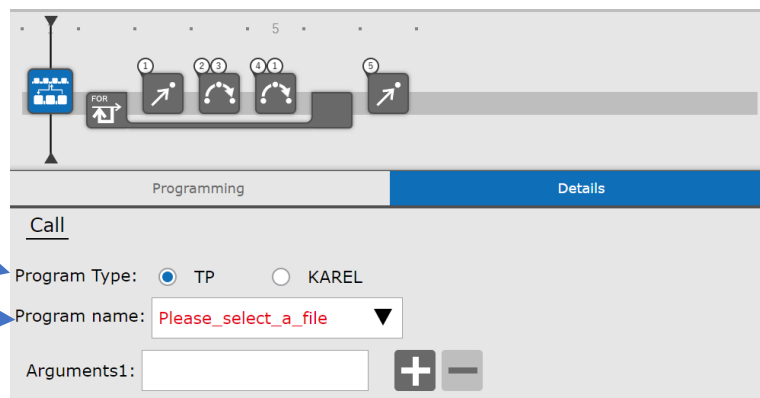
Drag and Drop Call Command to the front of the Program



- c. Select Call Command on the Timeline > Select Detail Tab

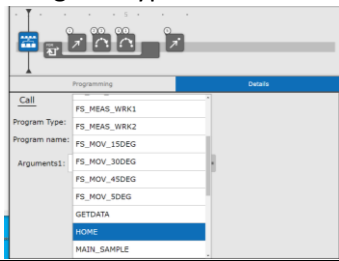
Program Type:  
TP: Programs created by teaching (moving) the robot to a desired set of coordinate points  
KAREL: Positional Structured Text Programming Language using registers and variables, similar to JAVA

Program Name: Allows User to Select program to call

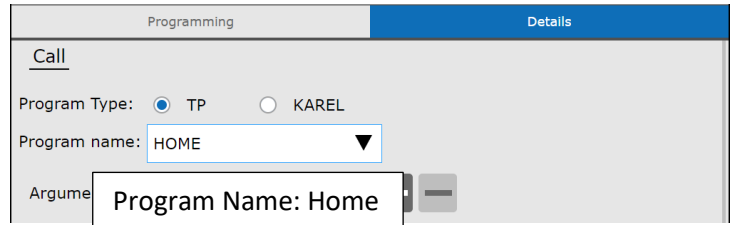




d. Select Program Type TP > Select Program Name > Select Home



Program Name > Select Home



e. Add Linear or Joint Command(s) that move the Robot Arms End Effector from the Home Position to about the center point of the circle

f. Run the Program in Teach Mode to ensure the End Effector does not collide with the table.

**Submission:** Shoot a Video of the Robot Running > Share Video with the Teacher

**Assignment 1:** Create a Program that will

1. Following the path of the curved line shown
2. Follow the Path Back to the Start Position Following the Arc
3. Loop the Path Forward and Backward 5 Times
4. After the Loop Set End Effect at the Center of the Arc Path
5. Call Home Program to Return Home
6. Add a Forever Loop to run the Program Continuously

NOTE: Robotic Arm's End Effector (Gripper) will be a minimum of 2 Inches from the table top

**NOTE:** Printout Provided at the Arm

