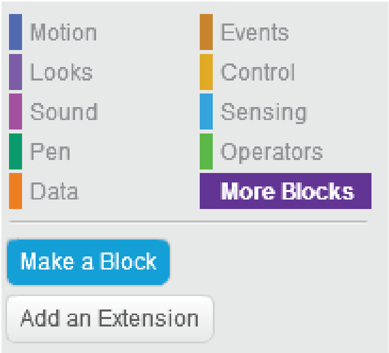
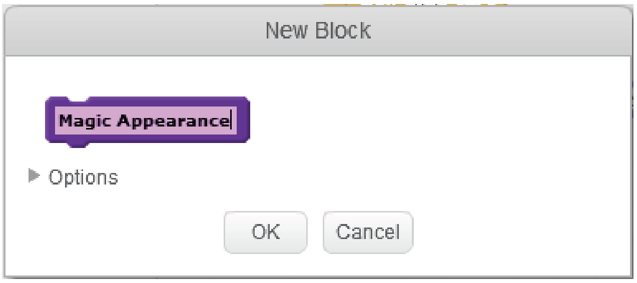
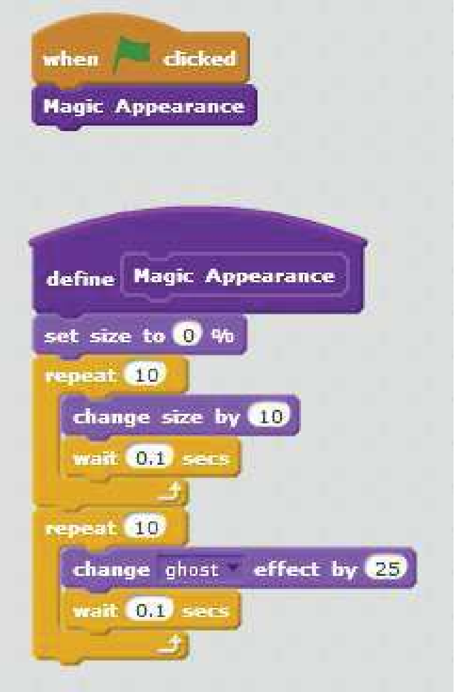
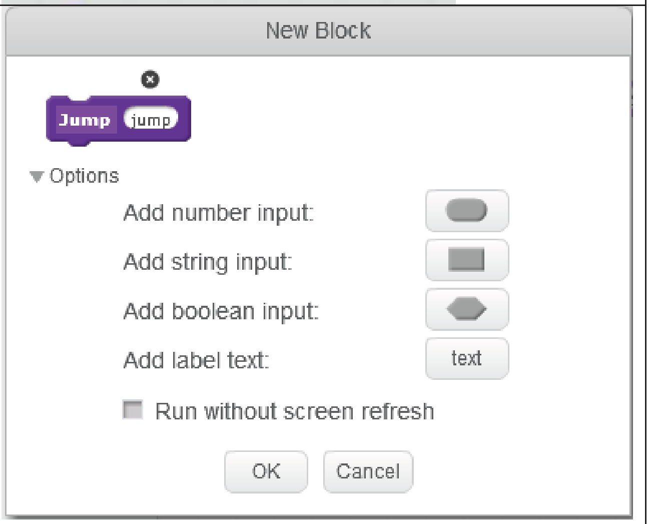
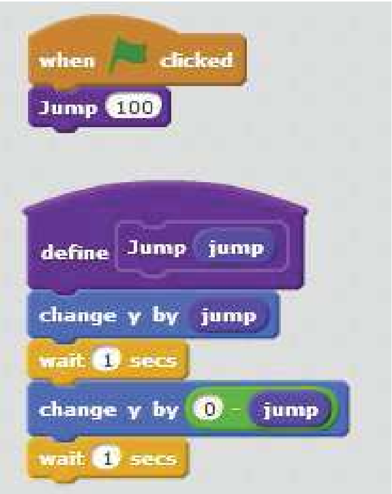
Creating your own blocks- Functions

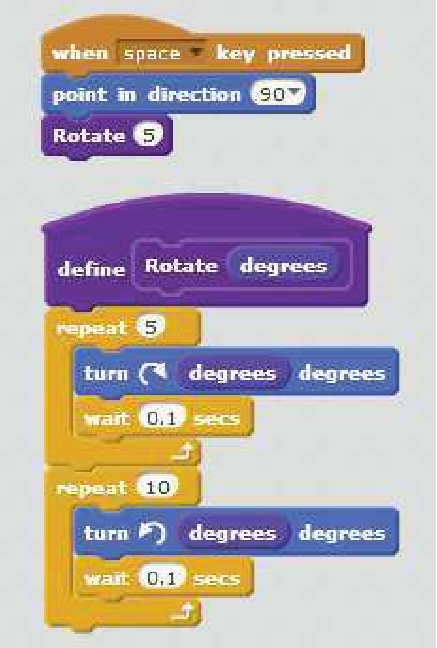
My Blocks (known as More Blocks in Scratch 2.0) is one of the ten categories of Scratch blocks. It holds procedures for the selected sprite. They are color-coded pink. Before any blocks are created, it is empty, except for a "Make a Block" button. Clicking Make a Block brings up a dialogue allowing the user to make a procedure. Once OK is pressed, the new block appears in the palette and an empty definition appears in the code area. When the procedure runs, Scratch will run the blocks below the corresponding Define block.

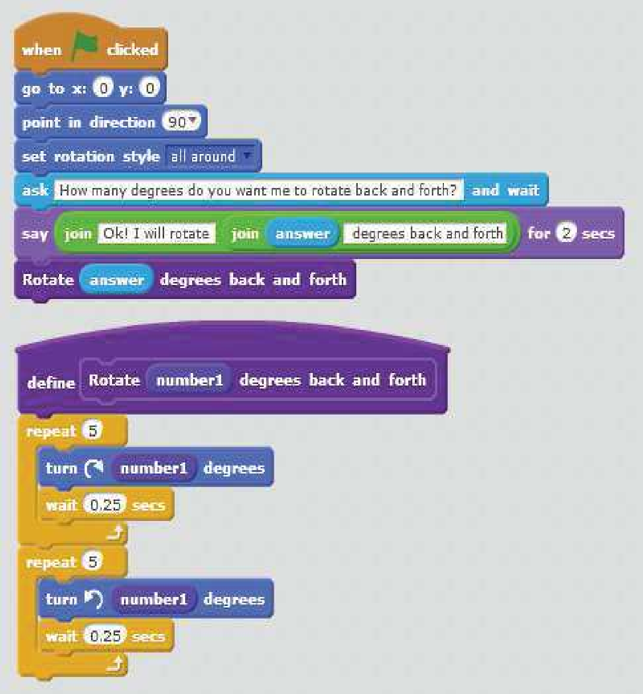
Uses of My Blocks

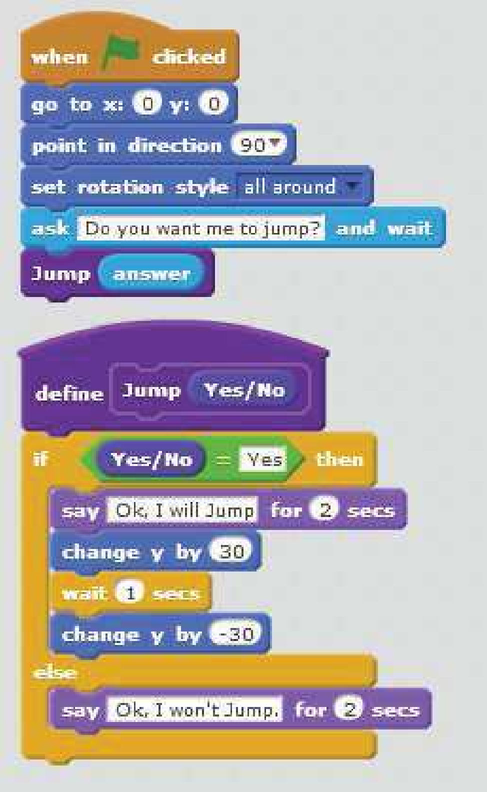
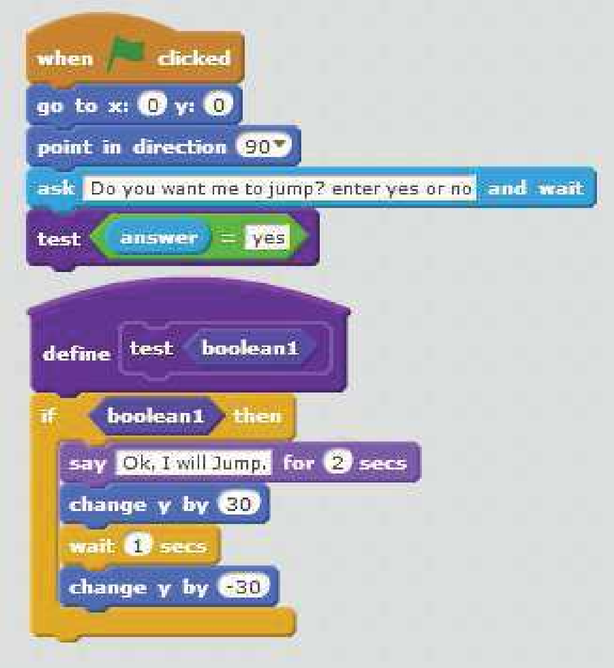
'Make a Block' provides the ability to reuse the same code in multiple places without having to duplicate the code. It also allows one part of code to pass information to another part. Some programmers call reusable code that is called by another part of a program a 'subroutine' or 'function'.

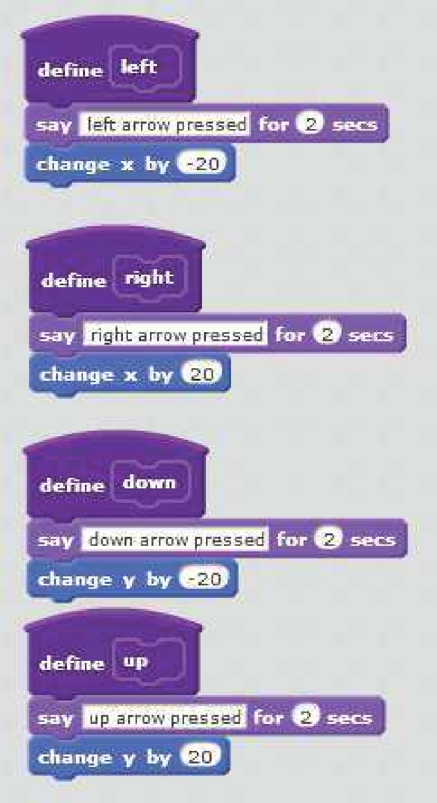
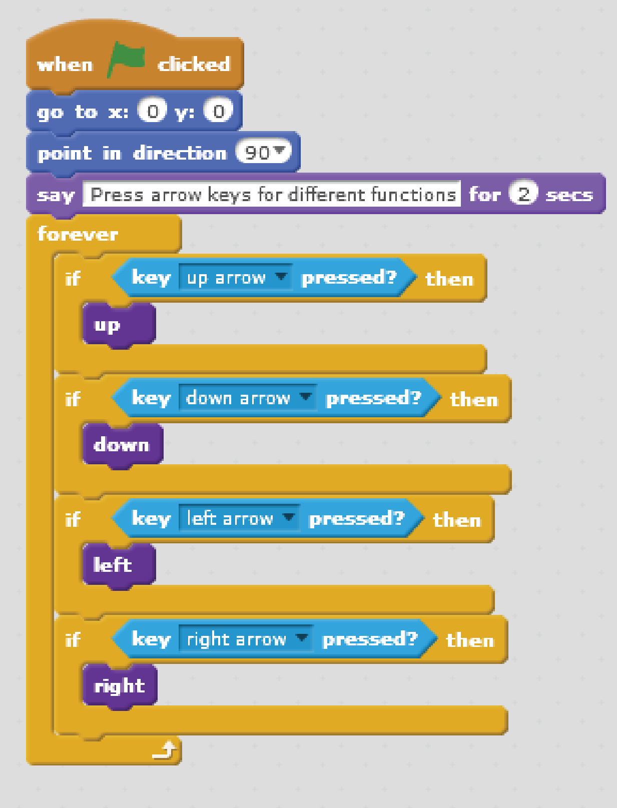
Sl. No.	Activity Name	Activity Image
1	Create a new block i) To create a new block in Scratch , you need to go to the block palette and select the **More Blocks** category. ii) Once here, you click the **Make a Block** button to open the **New Block window** iii) Here you **enter a name for the Block** and select options for parameters passing. iv) Click the OK button when ready. v) After you create a new block, we have to add code under the define *Block* area.	 Motion Events Looks Control Sound Sensing Pen Operators Data More Blocks Make a Block Add an Extension  Magic Appearance

		 when 🏁 clicked Magic Appearance define Magic Appearance set size to 0 % repeat 10 change size by 10 wait 0.1 secs repeat 10 change ghost effect by 25 wait 0.1 secs
2	Passing parameters to new block	 New Block Jump jump Options Add number input: Add string input: Add boolean input: Add label text: text □ Run without screen refresh OK Cancel  when 🏁 clicked Jump 100 define Jump jump change y by jump wait 1 secs change y by 0 - jump wait 1 secs

3	Rotate Function	
		when space key pressed
		point in direction 90
		Rotate 5
		
		define Rotate degrees
		repeat 5
		    turn ↻ degrees degrees
		    wait 0.1 secs
		repeat 10
		    turn ↺ degrees degrees
		    wait 0.1 secs

4	Rotate number input	 when 🏴 clicked go to x: 0 y: 0 point in direction 90 set rotation style all around ask How many degrees do you want me to rotate back and forth? and wait say join Ok! I will rotate join answer degrees back and forth for 2 secs Rotate answer degrees back and forth define Rotate number1 degrees back and forth repeat 5 turn ↻ number1 degrees wait 0.25 secs repeat 5 turn ↺ number1 degrees wait 0.25 secs

5	Jump String Input	 when clicked go to x: 0 y: 0 point in direction 90▾ set rotation style all around ▾ ask Do you want me to jump? and wait Jump answer define Jump Yes/No if Yes/No = Yes then say Ok, I will Jump for 2 secs change y by 30 wait 1 secs change y by -30 else say Ok, I won't Jump. for 2 secs
6	Jump Boolean Input	 when clicked go to x: 0 y: 0 point in direction 90▾ ask Do you want me to jump? enter yes or no and wait test answer = yes define test boolean1 if boolean1 then say Ok, I will Jump. for 2 secs change y by 30 wait 1 secs change y by -30

7	Sprite moving with keys using functions. 	

```
define left
say [left arrow pressed] for (2) secs
change x by (-20)

define right
say [right arrow pressed] for (2) secs
change x by (20)

define down
say [down arrow pressed] for (2) secs
change y by (-20)

define up
say [up arrow pressed] for (2) secs
change y by (20)
```

```
when ⚑ clicked
go to x: (0) y: (0)
point in direction (90▼)
say [Press arrow keys for different functions] for (2) secs
forever
    if <key [up arrow ▼] pressed?> then
        up
    if <key [down arrow ▼] pressed?> then
        down
    if <key [left arrow ▼] pressed?> then
        left
    if <key [right arrow ▼] pressed?> then
        right
```