05 Programme Development

5.1 Decomposing the Problems

A problem can be simple or complex according to its nature. It is rather difficult to perceive and understand a complex problem when compared with a simple problem. It is essential to understand a problem thoroughly before going to solve it. So it will be easier to find a solution to a complex problem after decomposing it into smaller sub-problems.

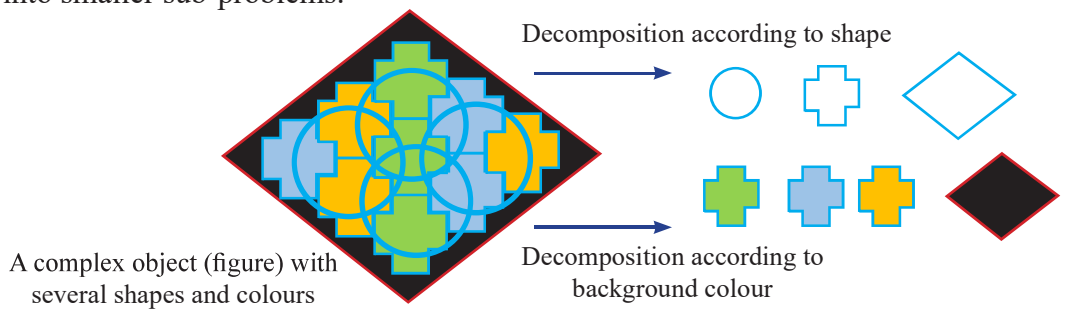
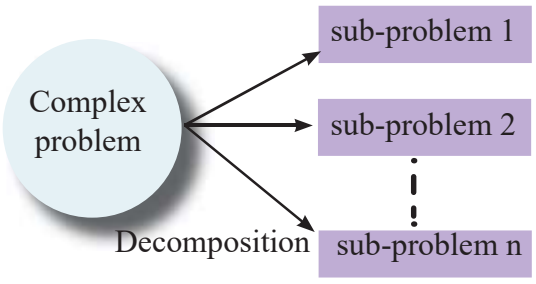


Figure 5.1 - Decomposition of a complex object

Likewise, a complex problem that seems difficult to solve at once can be broken into several smaller sub-problems as much as possible. Then these sub-problems can be solved easily one by one.

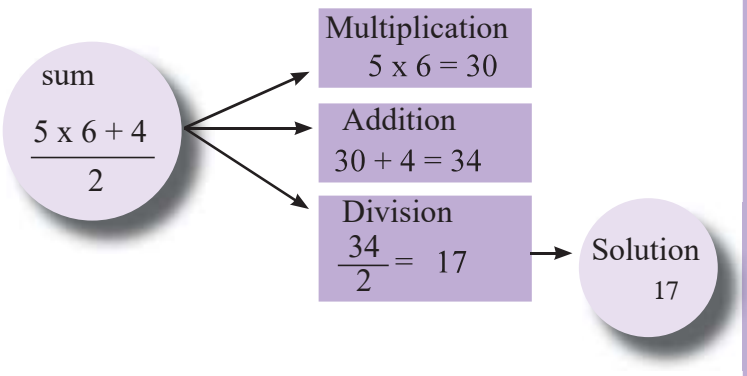


Advantages of decomposing a problems

- Easy to solve
- Easy to understant
- Easy to describe
- Easy to analyze

Figure 5.2 - Decomposition of a complex problem into sub-problems

eg:- In grade o6, you have learnt about the basic mathematical function which include addition, substraction, multiplication and division of a whole number. Additionally in the first term of graded 07, you have learnt about solving sums when more than one mathematical functions are used with a whole number.
Consider the following sum (Refer Figure 5.3).



Since it is hard and complicated to solve this sum at once (in one step), it would be simple and easy to find a solution after decomposing the sum into three sub-problems as multiplication, addition and division involved in the sum. The final solution can be reached easily by logically integrating the answers obtained in each sub-problem.

Figure 5.3 - Breaking a mathematical sum into sub-problems

Activity 1 : See Workbook 5.1

5.2 Use of Flow Chart to represent Algorithm

Symbols in flow chart to represent algorithm have been explained in grade six. An algorithm may include one or more control structures out of three. Three types of control structures are shown below.

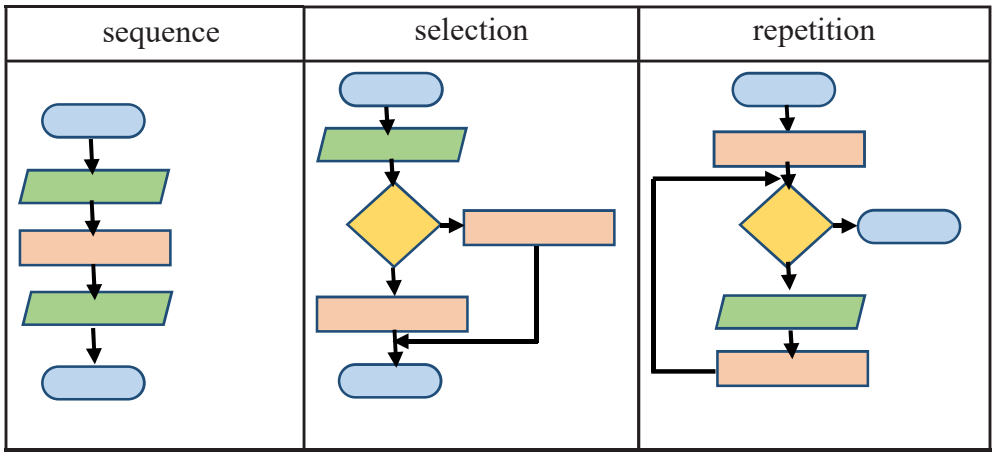


Figure 5.4 – Types of control structures shown in flow chart

Activity 2 : See Workbook 5.2

5.2.1 Sequence

Execution of instructions in an algorithm sequentially from top to the bottom is called sequence. Let's see how it is represented in a flow chart.

e.g. : Covering a textbook

Let's represent covering a textbook in a flow chart.

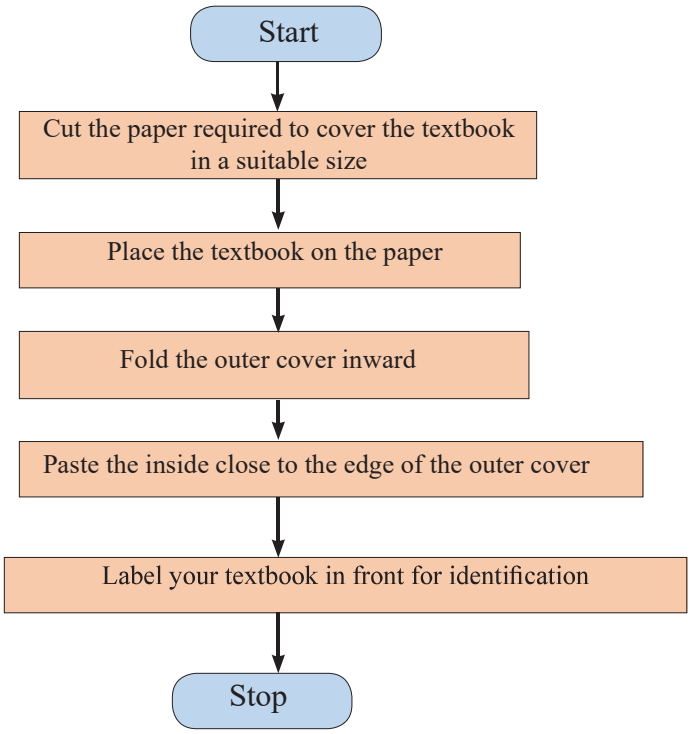
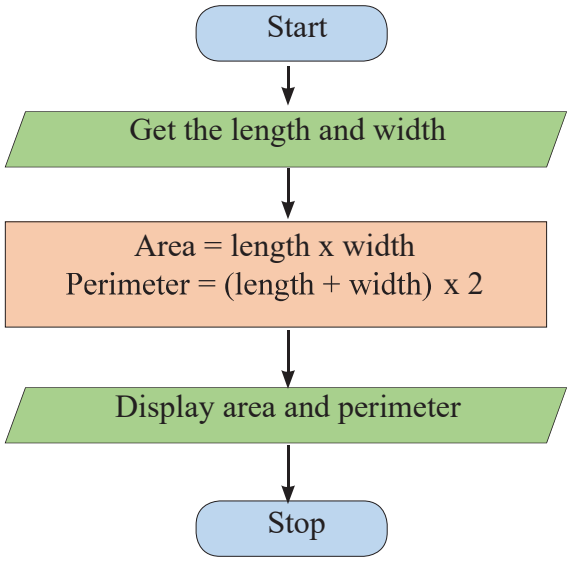


Figure 5.5 - Flow chart: Covering a textbook

Activity 3 : See Workbook 5.3

e.g. 2 :Finding the area and perimeter of a rectangle



Problem Analysis
Input : length and width
Process :
Perimeter = (length + width) x 2
Area = length x width
Output : area, perimeter

Figure 5.6 – Flow chart: Finding area and perimeter of a rectangle

Activity 4 : See Workbook 5.4

5.2.2 Selection

Here it is expected to make a making decision on which step to follow depending on the condition given by the algorithm. In a selection, the condition is checked first and the flow direction is chosen based on whether the condition is true or false.

For example, let's consider an instance where a ceiling fan is operated. When the switch is on, the ceiling fan operates if there is power. If there is no power, the ceiling fan does not operate.

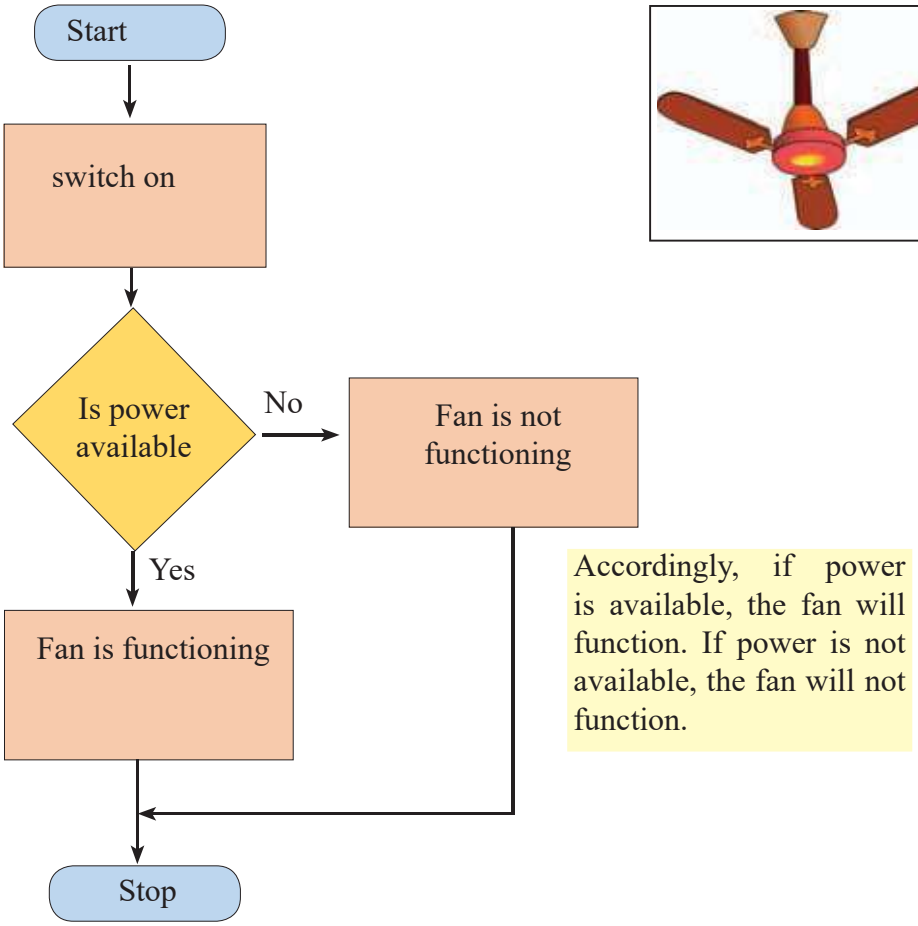
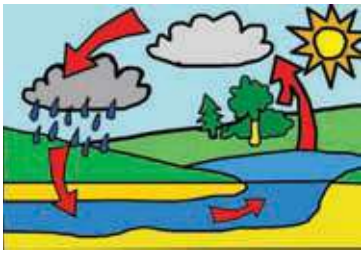


Figure 5.7 - Flow chart funtioning of a font

5.2.3 Repetition



Execution of an instruction or several instructions in an algorithm repeatedly until a condition is satisfied is called repetition. A repetition will take place depending on whether a condition is satisfied or not satisfied. For example, natural water cycle is a process that takes place repeatedly.

As an example, do you know that since your childhood, you used to save moneyby using a till to put money? The one who is used to save money in a till, will always put money (again and again) repeatedly until the till fills.So, here the process of putting money is repeated until the condition that is 'till becomes full' is satisfied.

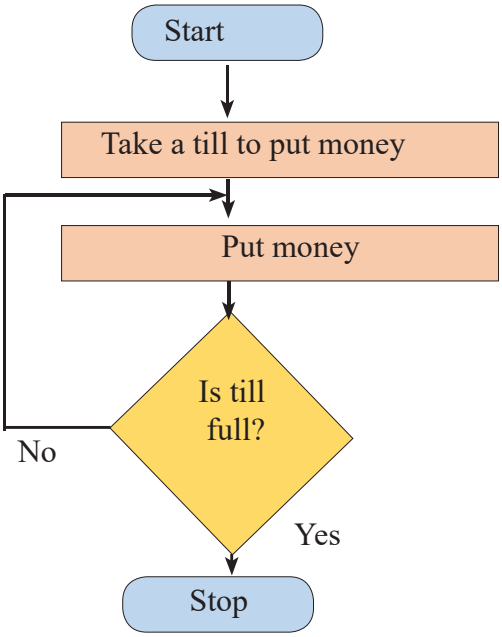


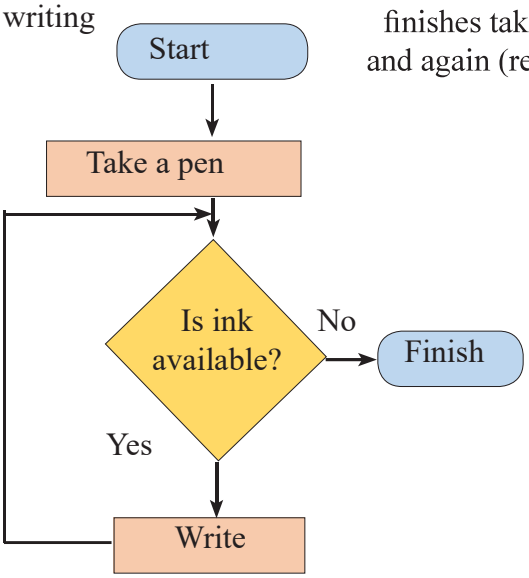
Figure 5.8 - Flow chart: Putting money into a till

Checking of condition for repetition can take place in two ways.
1. Check the condition before repetion starts.
2. Check the condition after functioning once.

Checking whether ink is available before writing

act of writing until ink finishes takes place again and again (repeated)

check whether ink is finished every time after writing



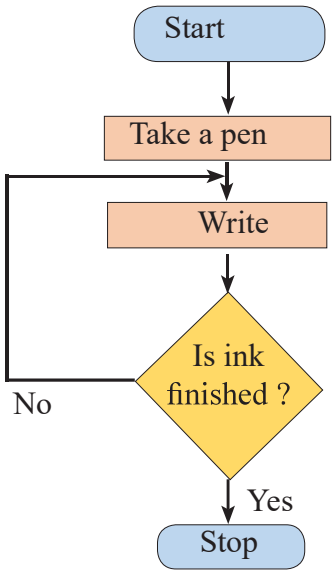
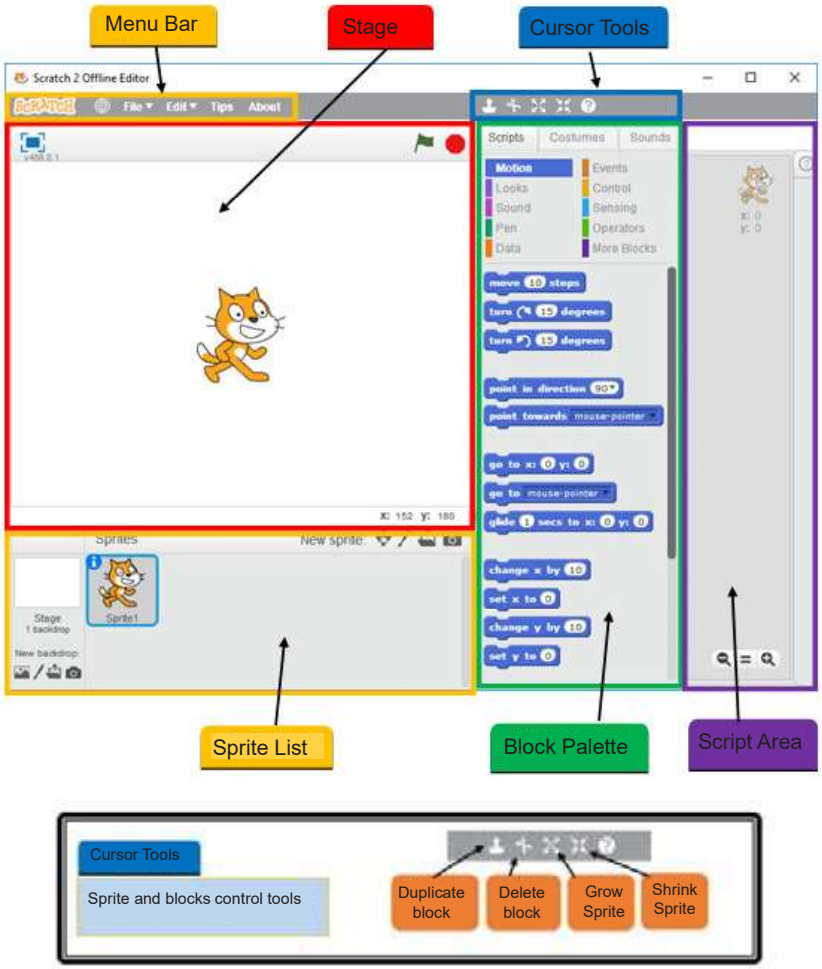


Figure 5.9 - Flow chart: Writing with a pen until ink finishes

5.3 Introduction to Scratch Programming and Programme Development

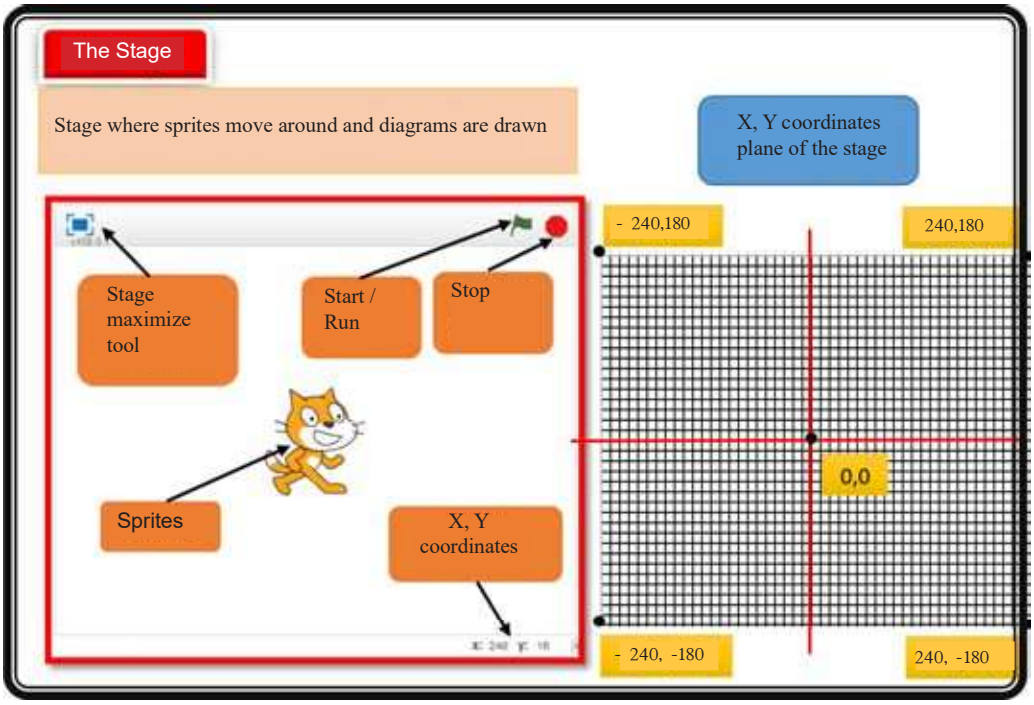
Scratch which is a visual programme development software is an interactive, attractive and a simple programme with command blocks. This is a Free and Open Source Software (FOSS) and therefore it can be downloaded freely from the following website. http://www.scratch.mit.edu

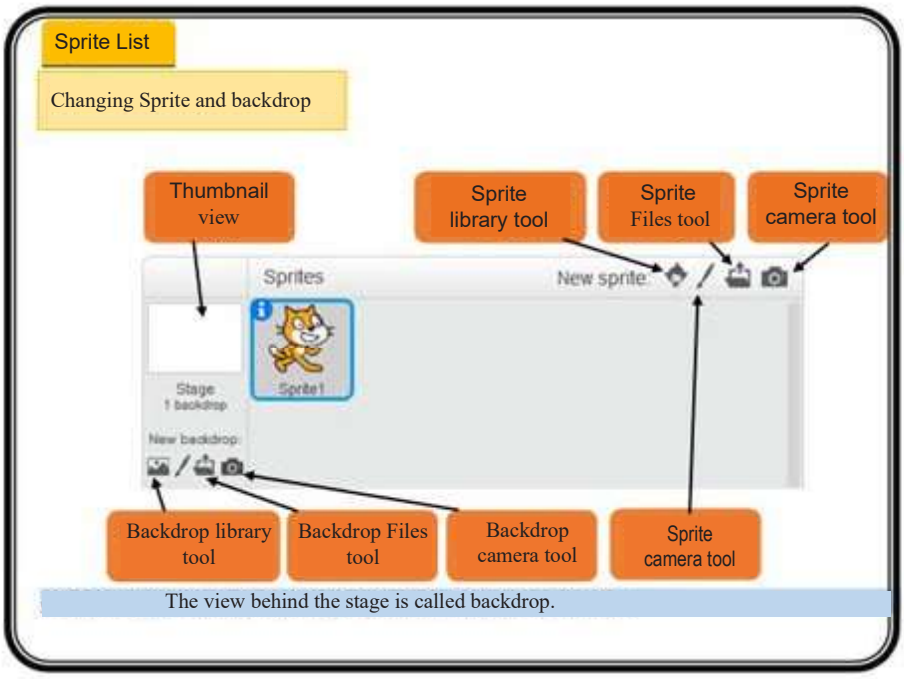
5.3.1 Scratch 2.0 introducing Interface with Visual Development Environment



Sprite and blocks control tools

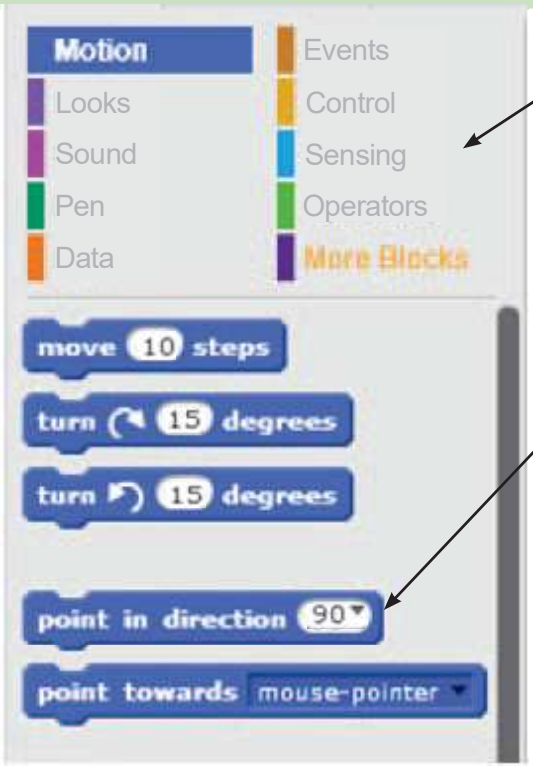
Note: All these interfaces are in Scratch 2.0 (version 2).


The Stage
Stage where sprites move around and diagrams are drawn
X, Y coordinates plane of the stage
Stage maximize tool
Start / Run
Stop
Sprites
X, Y coordinates
- 240,180
240,180
0,0
- 240, -180
240, -180


Sprite List
Changing Sprite and backdrop
Thumbnail view
Sprite library tool
Sprite Files tool
Sprite camera tool
Sprites
New sprite:
Sprite1
Stage
1 backdrop
New backdrop:
Backdrop library tool
Backdrop Files tool
Backdrop camera tool
Sprite camera tool
The view behind the stage is called backdrop.

Blocks palette

Block and the classified tab within it



Blocks Tabs

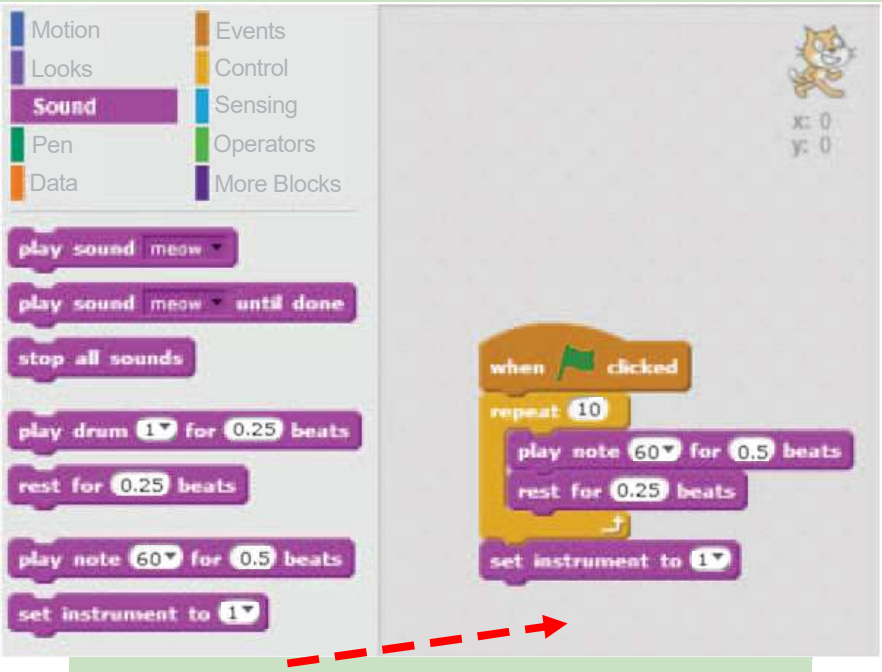
The blocks categories are all colour coded and the relevant block is displayed under the relevant tab.

Blocks

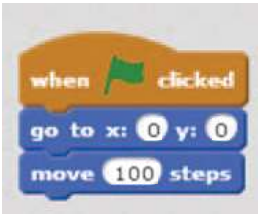
A list of blocks that shows the action on the stage is displayed.

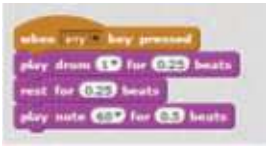
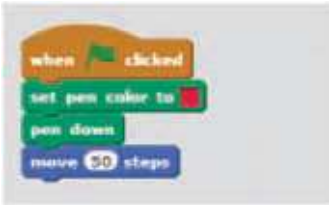
Script Area

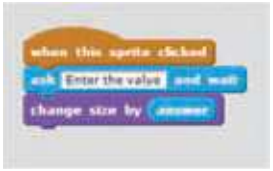
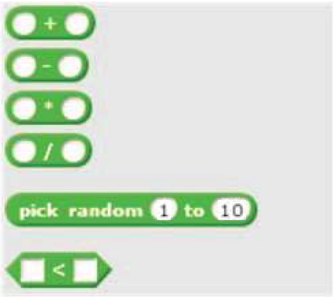
This is the area where you apply blocks and create (develop) programmes.



Creating a programme by dragging a block onto the ScriptsArea

Block classification	Block	Example
Motion Includes command blocks pertaining to move the sprite on the stage.		 Sprite moves 100 steps from the position 0, 0
Looks Includes command blocks pertaining to communication among the Sprites.		 Sprite, after saying "Hello!, Can you win" thinks "No, I'm the winner"

Sound Includes musical instruments and basic notes to create sound and sound patterns		 When any key pressed drum sound is played and then “s” note is played
Pen Includes colour and tools needed to draw lines and various shapes on the stage.		 draws a line 50 units long in red colour
Data Includes command block pertaining to make a variable and assign value.		 Count variable value is changed by 1 and then the value is shown
Events Give instructions pertaining to the execution (running) of all other blocks		 When run tool (green flag) clicked, the total of variables a and b is shown

Control Includes command blocks with selection and repetition to control execution of other blocks(scripts)		 According to the input, only one output out of the two is displayed.
Sensing Receive input		 size of the sprite changes
Operators handles mathematical operations		 display the total of numbers from 1 to 10

Activity 5 : See Workbook 5.5

5.3.2 Programs Development of Programmes

Follow the instructions given bellow to create a programme using scratch

- Run scratch software
- Double click on scratch icon

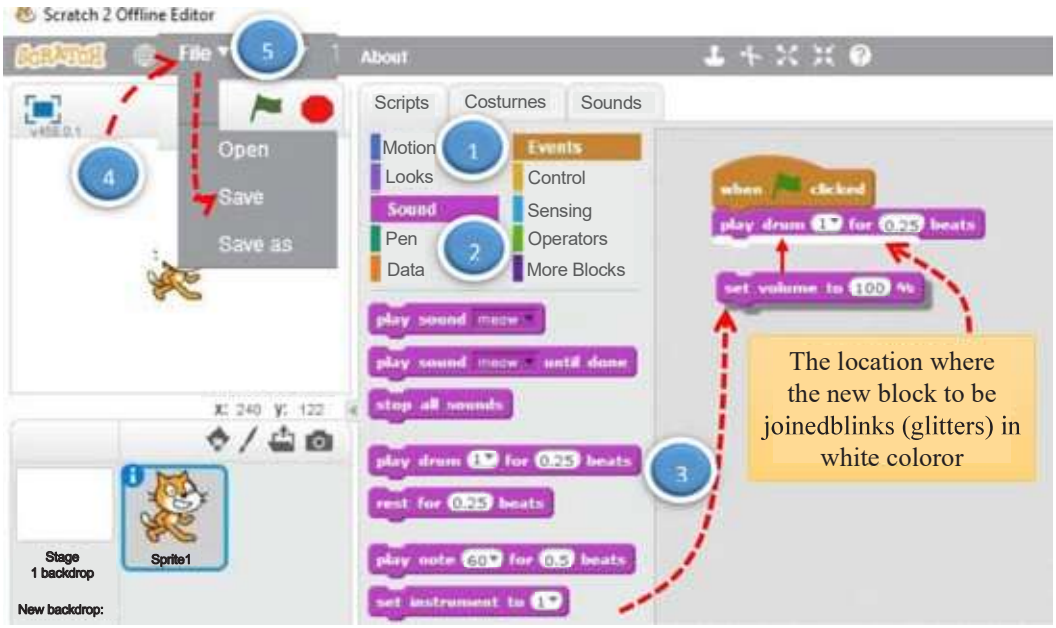
- Select types of command blocks under Script Tab
- Click on script tab and select block types

- Drag the command block and drop on Script Area
- Drag the blocks to script area

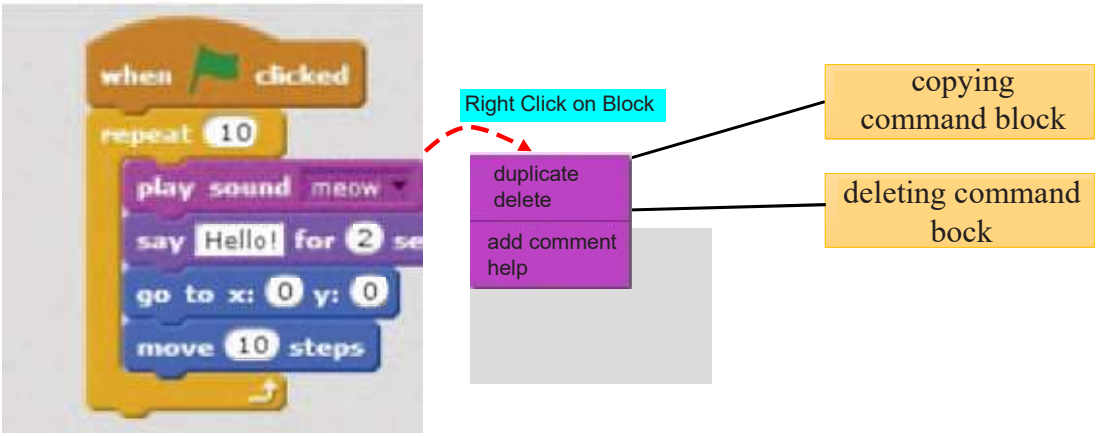
- Develop the programme

- Save as a file

- Run the programme



Managing Command Control Blocks



5.3.3 Development of programme with a Sequence

1. Playing basic notes with piano

Connect the control blocks (scripts) shown below sequentially. Then change the values of the control blocks as shown in the figure below. Run the programme and check the voice.

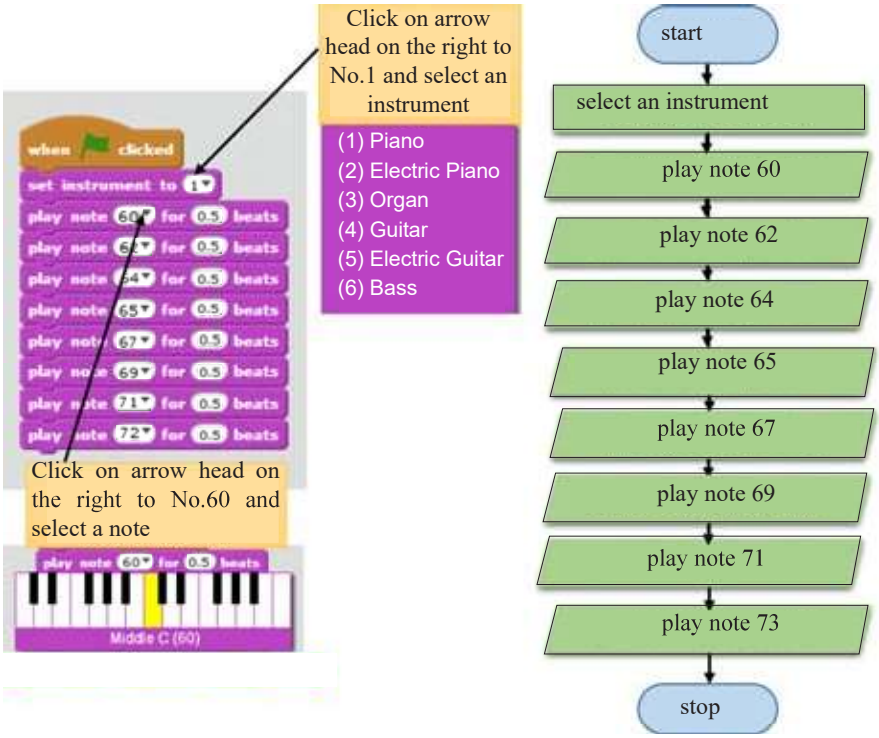
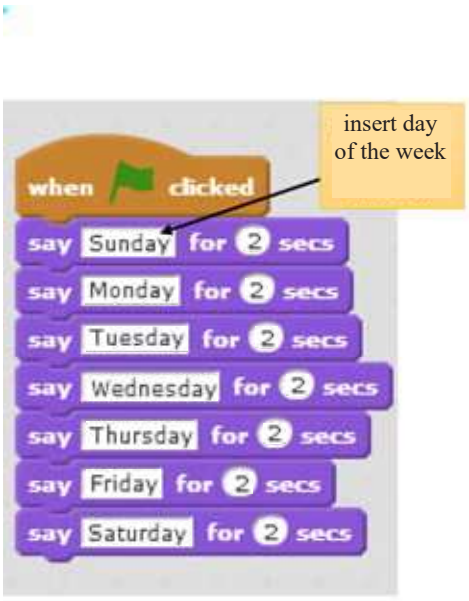
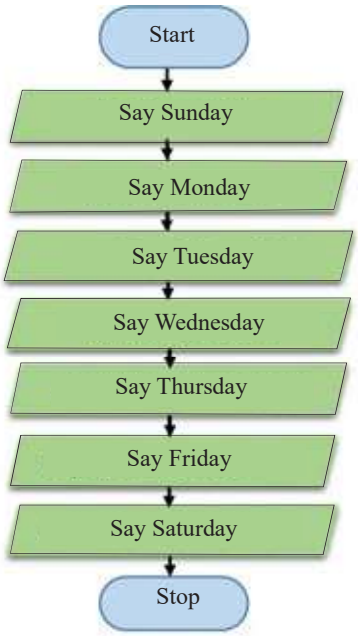


Figure 5.10 – Flow chart: Playing note

Activity 6 : See Workbook 5.6

2. Expressing days of the week in order by Sprite





5.4 Use of a variable

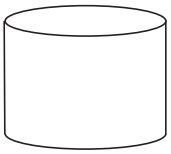
In programming, variables are used to store a value in memory temporarily. Let's study the following activity to understand the nature of a variable.

Let's consider a situation where two types of coloured liquid have been poured into two separate glass vessels.

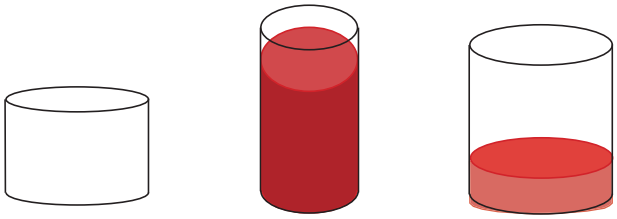
A vessel B vessel

Now let's consider changing liquid in A vessel into B vessel and liquid in B vessel into A vessel.

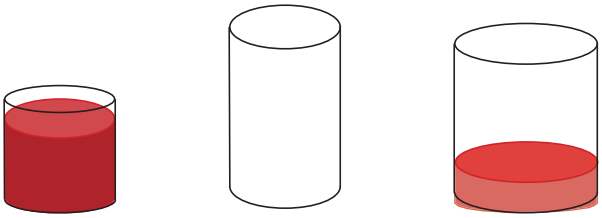
In this way, in order to change the liquid from one vessel to the other, it is necessary to have an extra vessel.
Let's name that vessel as C vessel.


C vessel

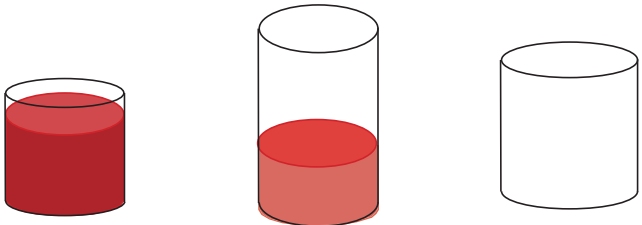
step 1 : Changing liquid in A vessel to C vessel


A vessel B vessel C vessel

step 2 : Changing liquid in B vessel to A vessel


A vessel B vessel C vessel

step 3 : Changing liquid in C vessel to B vessel

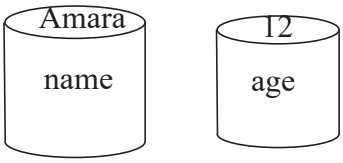

A vessel B vessel C vessel

As a vessel can store liquid, so a variable can store a value. The above vessels named as A, B and C can be considered as variables and the liquid poured into them can be considered as values of the variables.

Assign Values to Variables

When assigning values to variables, variable name is written on the left to the equal sign and the value is written on the right to the equal sign.

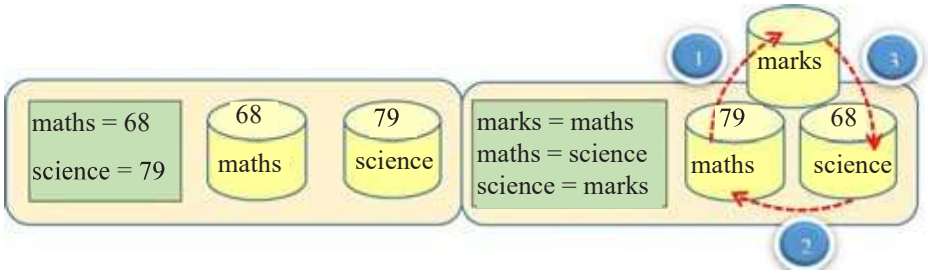
e.g. name = Amara
age = 12



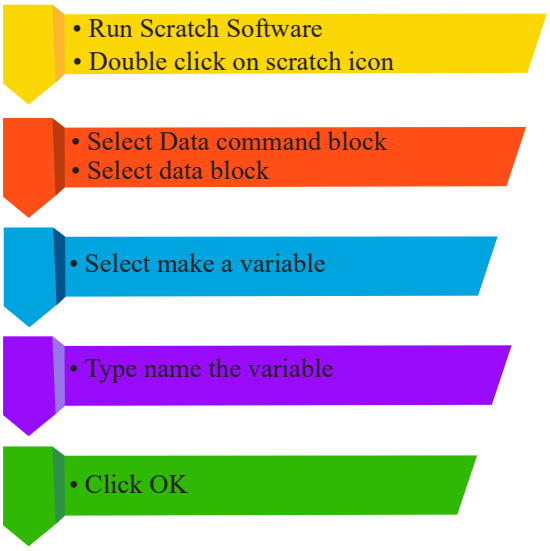
Changing values of variables

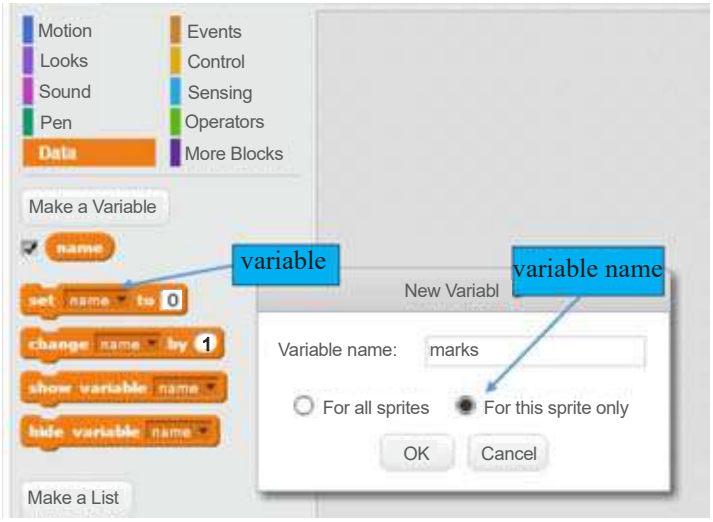
e.g.

Let's change the values of variables maths and science.



Building variable in scratch





assign value to a variable

(assign marks 67 to the variable marks)

change value of a variable

(change the value of the variable marks by 10)

5.4.1 Develop programme with variables

1. Multiplication of two numbers

Create two variables (n1, n2) to store values of two numbers and another variable to store the multipication of the two numbers.Then join the control blocks shown below one by one sequentially in the given order. Change the value of control blocks as shown.

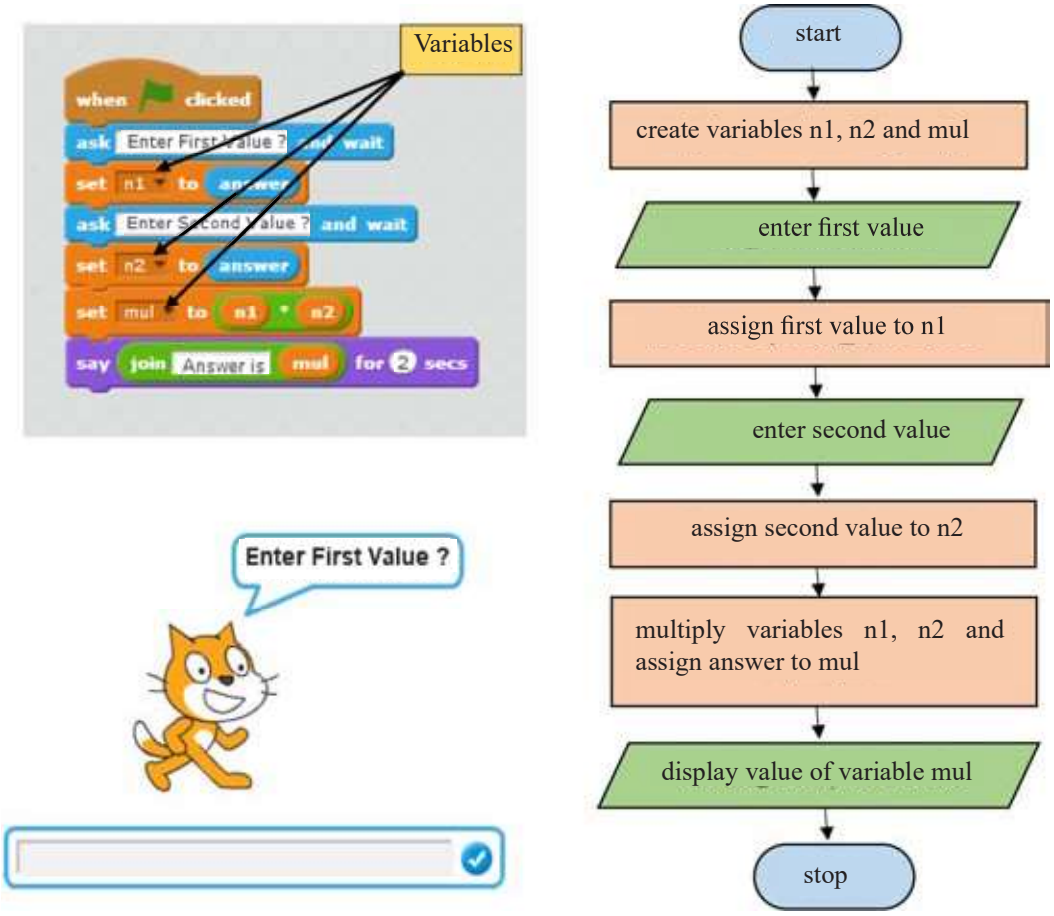


Figure 6.12 – Flow chart: Multiplication of two numbers

Activity 7 : See Workbook 5.7

2. Find Perimeter and Area of a Rectangle

The length and width of a rectangle is needed to find the perimeter and the area of that rectangle. Accordingly, four variables should be used in order to develop this programme. The length, width, perimeter and the area are shown as variables L, W, perimeter and area respectively.

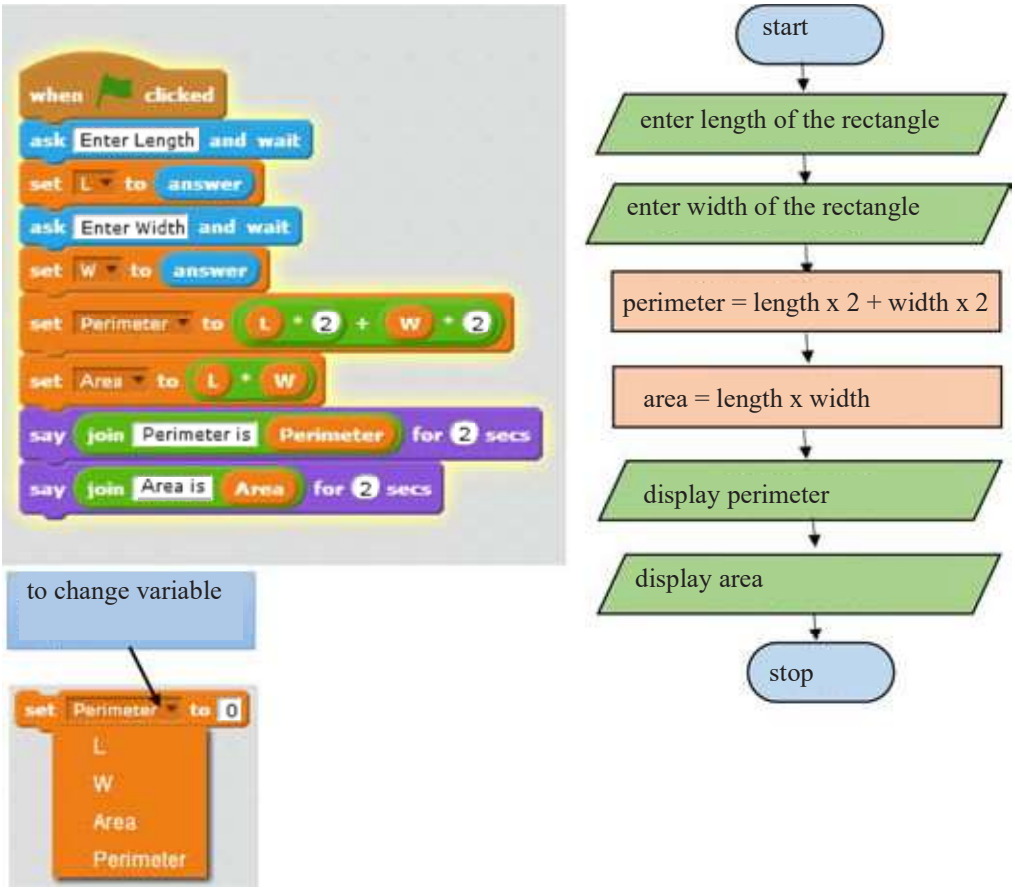


Figure 5.13 – Flow chart: Area and perimeter of a rectangle

3. Change the size of the Sprite from 10 to 100 at random

Develop the following programme to randomly change the normal size of the Sprite that we see. A value randomly chosen from one to ten is stored in the variable X. The size of the Sprite changes up to the value obtained by multiplying the value of variable X by 10. In addition, the colour too changes in proportion to the value of variable X.

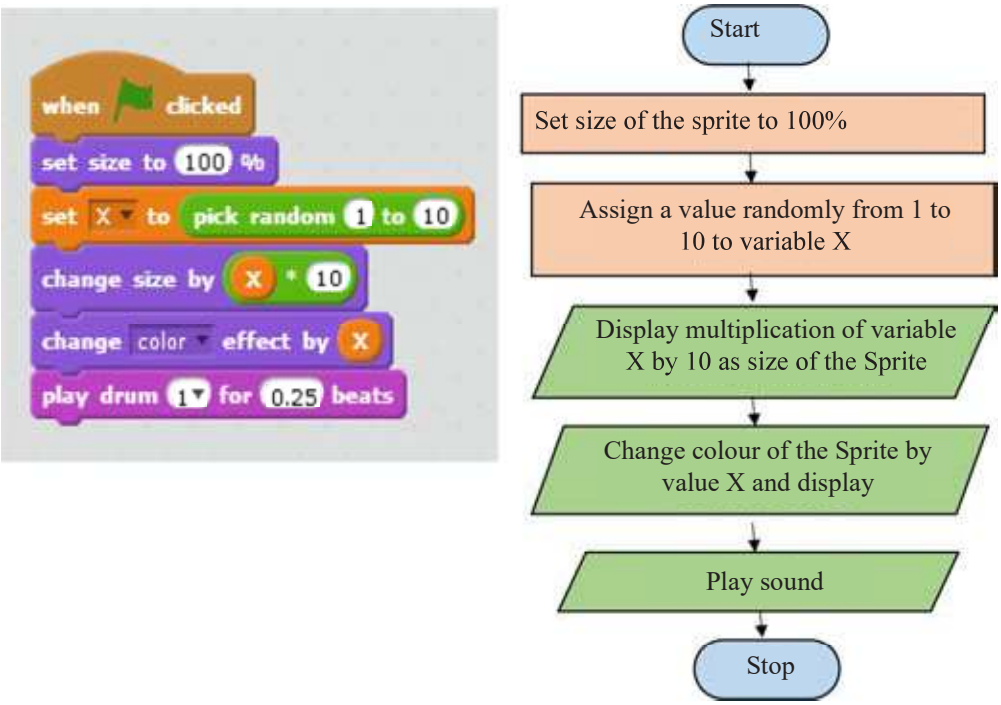


Figure 5.14 - Flowchart: Changing size of the Sprite

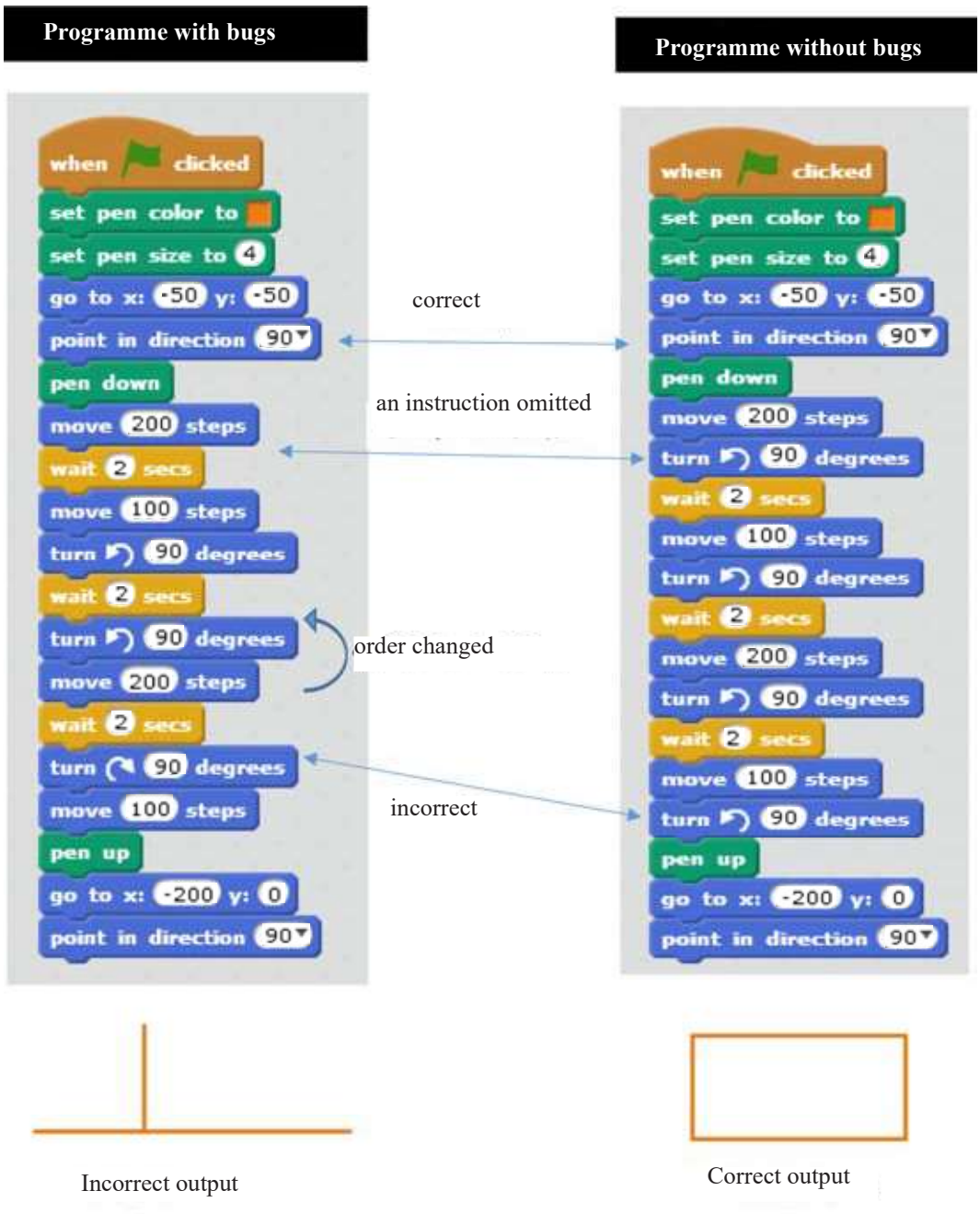
5.4.2 Bugs

Bugs may occur when developing a programme. Errors in a programme are called bugs, while the elimination of such bugs is called debug. In such a situation, unexpected problems are to be faced when a program with bugs run. Programmes with bugs may not run properly. Therefore before running a programme, bugs that may occur in that program should be eliminated.

Elimination of Bugs

Bugs may occur in such situations as change of order in the instructions (block), omission of instructions (block) or non-mentioning of correct variables and their values. Bugs in a programme can be eliminated by correcting bugs in the instructions(block) or arranging the order of the instructions (block) sequentially.

Let's compare the following programme with bugs and the programme without bugs developed to draw a rectangle with length and width 200 and 100 respectively.



Incorrect output

Correct output

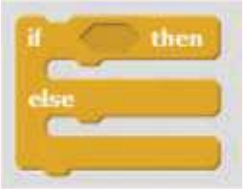
The above programme on the left has been developed to draw a rectangle. However the expected output cannot be achieved due to bugs in the programme. Therefore there is a possibility to get an erroneous output. It is proved by the programme on the right that a correct output can be achieved after eliminating those bugs.

A program with bugs may not run properly. After eliminating the bugs, the programme can run properly.

Summary

- Program development can be made easy by decomposing a complex problem
- There are three control structures used to develop a programme. They are sequence, selection and repetition.
- Execution of instructions (sequentially) step by step in an algorithm is called sequence.
- Making decision as to which step to follow based on the condition given by the algorithm is called selection.
- Flow direction in a flow chart is determined based on the condition of the flow chart.
- Scratch open source programme can be used for visual programme development.
- Command blocks are used in the development of Scratch program
- Following command blocks can be used to display the decision
  - The block to be used to show steps to follow only if the condition becomes true

  - The block to be used to show steps to follow if the condition becomes true or not



- In programming, variable is used to store value in memory.
- Error in a programme is called bug.
- Elimination of bug in a programme is called debug.