

## **Unit I: Computational Thinking and Visual Programming Concepts**

Introduction to computational thinking. Visual programming concepts. Scratch environment: sprites -- appearance and motion, angles and directions, repetition, and variation, changing costumes, adding background. Input/Output, variables, and operators.

### **1. Elaborate on Computational Thinking(CT)**

#### **Computational Thinking(CT):**

- Computational thinking is **learning to think like a computer**
  - **Computational thinking** is the ability to **objectively analyze and evaluate a complex problem, develop possible solutions to the problem, then format these solutions in a way that a computer – or a human – could understand and carry out.**
  - **Core steps to computational thinking:**
    - **Decomposing** : Take a large and complex problem, and **break it down into** smaller, more manageable problems or steps
    - **Pattern Recognition**: Look for patterns or trends among the problems .Identify or **generate data which is useful to the problems.**
    - **Abstraction**: **Remove unnecessary detail** from the problem
    - **Algorithms** : Once the problem is broken down accordingly, **take those pieces, solve them individually**, and put them together logically into a complete solution.
    - **Analyze and evaluate the solution** :What **new insights** do you have based on the solutions to your problem?
-

## **2.What is Visual Programming:**

A visual programming language (VPL) is a programming language that uses graphical elements and figures to develop a program.

Kodu, Blockly, executable UML ,Scratch, Raptor are popular examples of visual programming languages.

Advantages of VPL

Fewer programming concepts,, Concreteness, Explicit depiction of relationships ,Eliminates an intermediate step in the process of creating a program.

## **3.What is Scratch**

Scratch is a free programming language and online community where you can create your own interactive stories, games, and animations.

- Scratch is a block-based visual programming language. •

Block programming uses graphical blocks to represent programming commands eliminating typing and syntax errors.

- Scratch is a good way for people of all ages to enter the world of programming.
- It provides an interactive environment to create programs by dragging blocks of code

Scratch 3.0—latest version



#### **4) Defining the following terms:**

##### **a) Stage**

The stage is basically the background of your project. Stage is the area where the sprites are and perform their actions.

Like sprites, the stage can have different costumes that change as the story plays out.

Stage is at upper right of code area (where actors act and information is displayed).

Stage area is 480\*360 pixels

##### **b) Sprite:**

An active object(actor) in scratch is often referred to as sprite that perform actions in a project.

Sprites can be programmed using instructions(script) in scratch interface to control them.

Every Scratch program is made up of sprites and the scripts (instructions) .

##### **c) The script**

A script is another name for program or method;

scripts are usually referred to as sets of blocks that consist of at least two blocks.

a script tells the actor what to do.Script is written in script area

#### **d) The programming palette/block palette**

The block palette is where the different script blocks are located.

The various Code Blocks are: • Motion • Looks • Sound • Events • Control  
• Sensing • Operators • Variables • My blocks • Extensions Programming

e) panel in center; here is where we construct scripts for the sprites. Blocks can be dragged from the Block Palette into the Code Area and arranged to form scripts.

Green flag is used to start the program. • Red flag is used to end the program.

**Code Area (Scratch)** The Code Area is the large empty space to the right of the Block Palette. It is an area for storing blocks that run the project. Blocks can be dragged from the Block Palette into the Code Area and arranged to form scripts.

**A costume** is another picture of the same sprite that is just slightly different

#### **5) List the various code blocks in scratch and elaborate on any 4.**

The various Code Blocks are:

- Motion
- Looks
- Sound
- Events
- Control
- Sensing
- Operators
- Variables
- My blocks
- Extensions Programming

## a) Motion blocks (18)

- They are color-coded medium-blue and are used to **control a sprite's movement.**
- controls the up, down, left, right, and rotation movement of the sprites.
- There are currently 18 Motion blocks: 15 stack blocks and 3 reporter blocks.
- A few Motion blocks are:
  - **Move 10 steps:** move sprite by a certain number of steps
  - **turn clockwise 15 degrees, turn anticlockwise 20 degrees:** rotate a sprite in a certain direction to a specified degree
  - **goto random position, goto x,y,**
  - **glide 1 second to random position:** used to change the rate at which the sprite moves
  - **point in direction 90 degrees, point towards mouse pointer,**
  - **change x by 10, change y by 10,**
  - **set x to 0, set y to 0**
  - **if on edge bounce:** tells a sprite to bounce off the screen if it is touching the edge of the screen., set rotation style all around, display x position, y position, direction.

## b) Event blocks (8)

- They are color-coded light yellow
- **used to sense events, trigger scripts to run.**
- There are currently 8 Events blocks: 6 Hat blocks (these blocks are used to start scripts) and 2 Stack blocks. It is the smallest category of blocks.
- **The following are the 8 event blocks:**
  - **When Flag Clicked:** is a hat block ,commonly called the **Start Block**
  - **When space/up arrow,down arrow,right arrow,left arrow,any, a-z, 0-9 key pressed**
  - **When this sprite clicked:** Scripts that wear the block will activate once its sprite or clone of the sprite is clicked.
  - **When backdrop switches to backdrop1:** Scripts that wear this block will be triggered once the specified backdrop has been switched to on the Stage

- **When loudness/timer >10**
- **When I receive message1:** The block activates its script when the specified broadcast has been sent by a calling script.
- **Broadcast message1----stack block :** which sends a broadcast throughout the whole Scratch program.
- **Broadcast message1 and wait-----stack block**  
**Broadcast message1 and wait-----stack block**

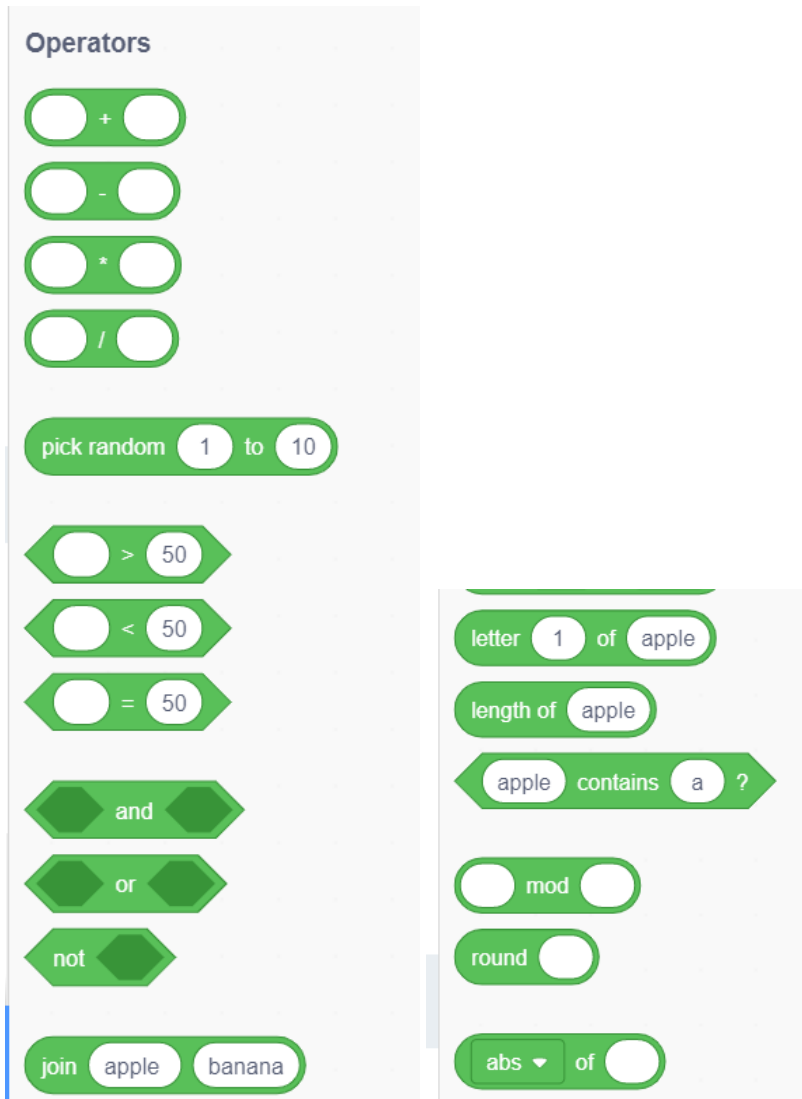
### c) Control block ( 11 )

- Control blocks are **color-coded gold and are used to control scripts.**
- allows you to specify things such as what starts and stops your sprites, how you can move the object, conditional operators such as if/then and repetition.
- There are 11 control blocks.
  - **Wait 10 seconds:** block pauses its script for the specified number of seconds
  - **Repeat 10 times**
  - **Forever**
  - **If condition is True then statements**
  - **If condition is true then**
    - **Statements**
    - Else**
    - **Statements**
  - **Wait until condition**
  - **Repeat until condition**
  - **Stop all**
  - **When I start as a clone**
  - **Create clone of myself**
  - **Delete this clone**

### d) Operator

Operators blocks is one of the ten categories of Scratch blocks. They are color-coded light-green and are used to script math equations and string handling.

The various operator blocks are:





## 6) Differentiate between sensing and looks block.

**Looks** - affects the appearance of your sprite such as color and costume.

**Sensing** - controls to sense if your sprite is touching the edge, another color, another sprite, at a specific X or Y coordinate, the sound volume, etc.

- **Looks block** are color-coded purple and are used to control a sprite's **appearance such as color and costume**.
- There are 20 looks blocks.
- A few looks blocks are:
  - **Say “hello” for 2 seconds, think Hmm for 2 seconds:** The block displays a speech bubble with the specified text for the sprite that runs it, which appears on the screen for the specified number of seconds.
  - **Switch costume to costume2**
  - **Next costume**
  - **Switch backdrop to backdrop1**
  - **Next backdrop**
  - **Change size by 10**
  - **Set size to 100%**
  - **Change color effect by 25**
  - **Clear graphics effect**
  - **show**
  - **Hide**
  - **Go to front layer**
  - **Display costume number,backdrop number,size**

## Sensing blocks

- are color-coded cyan and are used to detect different factors of a project.
- There are 18 sensing blocks
- **Function: controls to sense if your sprite is touching the edge, another color, another sprite, at a specific X or Y coordinate, the sound volume, etc.**
- The various sensing blocks are:
  - **Touching mousepointer/edge**
  - **Touching color**
  - **Color is touching**
  - **Ask “enter number” and wait**
  - **Answer**
  - **Mouse x**
  - **Mouse y**
  - **Display loudness**
  - **Display timer**
  - **Reset timer**

## 7) Explain the various operators available in scratch.

operators include tools to perform math functions, select a random number, greater than, less than, equal to, etc.

they are color-coded light-green, and are used to script math equations and string handling.

**Operator blocks can be used to perform mathematical operation(+,\*,/, -)**

- $() * ()$
- $() + ()$
- $() - ()$
- $() / ()$

**Used to compare values(<,>=) -----Returns true or false**

- `() = ()`

`() < ()` : [less than] block is an Operators block and a Boolean block. The block reports *true* if the first value is less than the second value and *false* otherwise.

- `() > ()`

### Used to perform logical operations(and, or, not)

- `() and ()` : The block joins two boolean blocks so they both have to be true to return *true*. If they are both true, the block returns *true*; if only one is true or none are true, it returns *false*.

X	Y	x and y	x or y
true	true	True	true
true	false	False	true
false	true	False	true
false	false	False	false

- `() or ()`
- `not()` -----`not(true)=false`

### On strings

`Join() ()` :

- The **Join ()()** [join] block is an Operators block and a Reporter block. The block concatenates, or "links" the two values together and reports the result — for example, if "hello" and "world" were put in the block, it would report "helloworld".
- Letter (index) of (string)
- `() Contains ()`----True or false output

### Math functions:inbuilt

- `() mod ()`
- `round()`
- `Abs/floor/ceiling/sqrt/sin/cos/tan/assign/log of()`

## 8) Explain the motions block in detail

- They are color-coded medium-blue and are used to **control a sprite's movement.**
- controls the up, down, left, right, and rotation movement of the sprites.
- There are currently 18 Motion blocks: 15 stack blocks and 3 reporter blocks.
- A few Motion blocks are:
  - **Move 10 steps:** move sprite by a certain number of steps
  - **turn clockwise 15 degrees, turn anticlockwise 20 degrees:** rotate a sprite in a certain direction to a specified degree
  - **goto random position, goto x,y,**
  - **glide 1 second to random position:** used to change the rate at which the sprite moves
  - **point in direction 90 degrees, point towards mouse pointer,**
  - **change x by 10, change y by 10,**
  - **set x to 0, set y to 0**
  - **if on edge bounce:** tells a sprite to bounce off the screen if it is touching the edge of the screen., set rotation style all around, display x position, y position, direction.

## 9) What are the different events that can occur in scratch

- **When Flag Clicked:** is a hat block ,commonly called the **Start Block**
- **When space/up arrow,down arrow,right arrow,left arrow,any, a-z, 0-9 key pressed**
- **When this sprite clicked:** Scripts that wear the block will activate once its sprite or clone of the sprite is clicked.
- **When backdrop switches to backdrop1:** Scripts that wear this block will be triggered once the specified backdrop has been switched to on the Stage
- **When loudness/timer >10**
- **When I receive message1:** The block activates its script when the specified broadcast has been sent by a calling script.
- **Broadcast message1----stack block :** which sends a broadcast throughout the whole Scratch program.
- **Broadcast message1 and wait-----stack block**

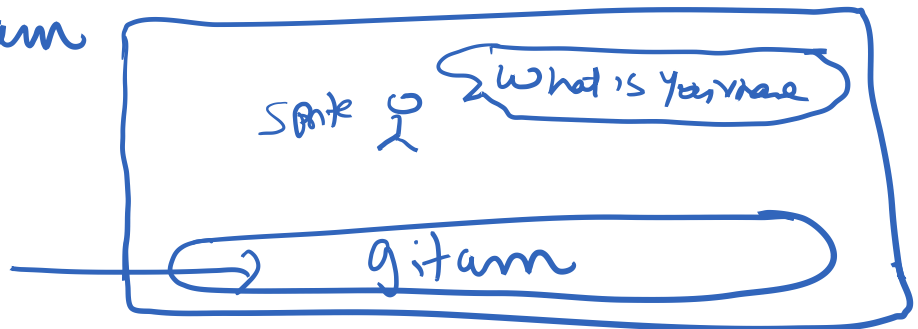
## 10) Explain how input is done in scratch

**In Sensing block :** The **Ask () and Wait** block is a Sensing block and a Stack block. The block will make the sprite using the block say the question and show an input box at the bottom of the screen. Scratchers can input text into it and submit it, and the input is stored then in the answer block. The **answer block** automatically updates to most recent input.

ask "what is Your name" and wait

answer = gitam

will be stored  
in answer



## 11) Define a variable? How value is assigned and changed for a variable in scratch

A **variable** is a changeable value recorded in Scratch's memory.

Variables can only hold one value at a time, unlike lists.

These values can be either numbers or strings,

their values can be input by the user via the answer block.

- There are two different types of variables, global and local
- Global variables can be altered and used by all sprites, whereas local variables can only be edited and used by the sprite they were created on

### **Value assigned to variable**

variables are created with the "**Make a Variable**" button in the **Variables Block Palette**. This results in only a small amount of RAM being used to store the value for use when the project actually runs.

After creating a variable use block **set myvariable to value**.

**Example: say we have created a variable 'a', then to assign value to 'a':**

**Set a to 10**

**Changing a variable value:**

**Change myvariable by (value)**

**Example: change a by 2**

**Output will be 12**

## 12) Illustrate the usage of my blocks in scratch

It holds **procedures** for the selected sprite.

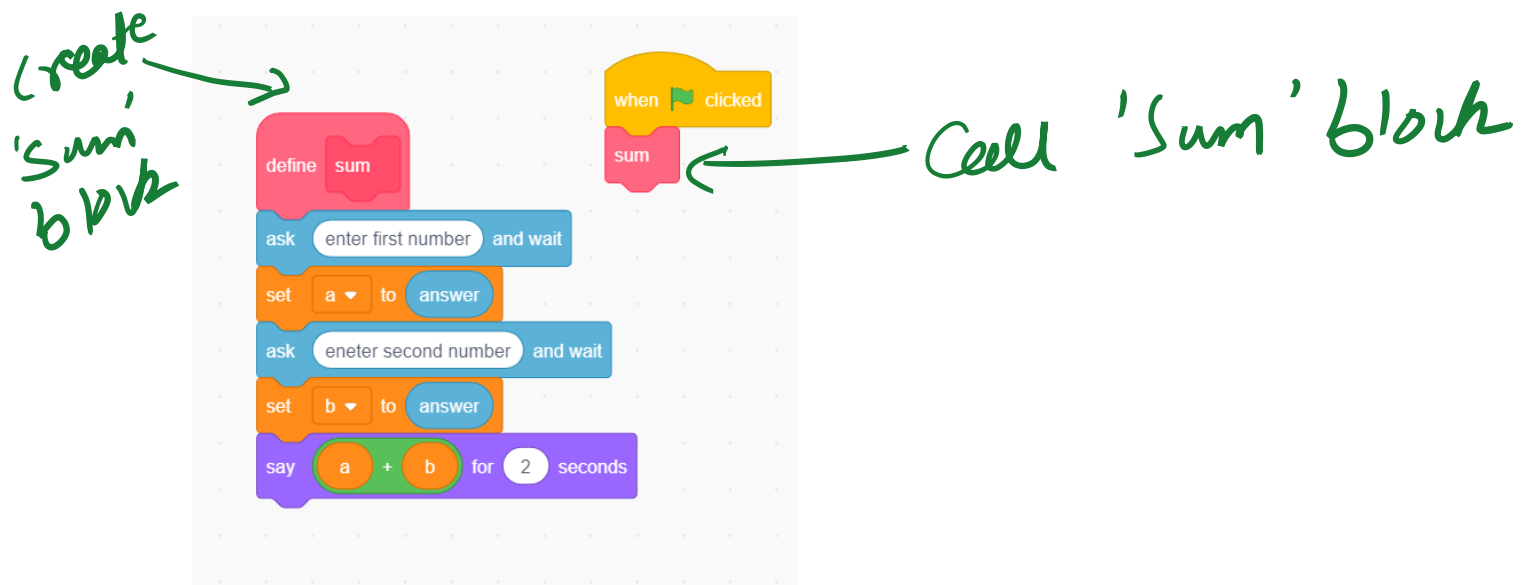
They are color-coded pink.

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.

- Myblocks is used for code reuseability and data abstraction.

Example



For example, suppose one wants a sprite to vanish and reappear. To vanish, the sprite should play a sound and repeatedly increase the ghost effect; to reappear, it should repeatedly decrease the ghost effect. This should happen at several points in the project. Without procedures, the Scratcher would have to duplicate the same sequence of blocks at each point, but by making "vanish" and "appear" blocks, they can write scripts in a way which matches how they think of them.



### 13) What are the options in pen block and explain how a geometrical shape is drawn in scratch?

The **pen** is a feature in Scratch that allows a sprite to draw shapes, animations, graphs, 3D projects.

**Options in pen block are:**

- Erase All — Clears all pen marks on the screen.
- Pen Down — Turns the pen feature on inside a sprite; the sprite will pen on the screen wherever it moves until the pen is turned off.
- Pen Up — Turns the pen feature off, stopping a sprite from penning.
- Set Pen Color to () — Sets the color of the pen to a predetermined color.
- Change Pen () by () — Changes a property of the pen by a given input.
- Set Pen () to () — Sets a property of the pen to a given input.
- Change Pen Size by () — Changes the size of the pen by a chosen number.
- Set Pen Size to () — Sets the size of the pen to a chosen number.
- Stamp — Draws a copy of the Sprite on the Stage.

### Geometrical shapes drawn in scratch(any 1 or 2 examples can be given)

#### square using loop

- 1.pen down
- 2.repeat 4
- 3.move 100 steps
4. turn 90 degrees
- 5.wait 1 second

#### rectangle using loop

- 1.pen down
- 2.repeat 4
- 3.move 50 steps

4. turn 90 degrees
5. wait 1 second
6. move 150 steps
7. turn 90 degrees
5. wait 1 second

#### **circle using loop**

1. erase all
2. pen down
3. repeat 360
4. move 1 step
5. turn 1 degree

#### **Draw Triangle**

1. erase all
2. Pen down
3. Repeat 3
4. move 50 steps
5. turn 120 degrees

#### **14) Explain repeat and forever blocks with example**

The **Repeat () block** is a Control block and a C block [A **C block** is a block that is shaped like a "C", so other blocks can fit inside it. These blocks perform the conditions and loops].

Blocks held inside this block **will loop a given number of times**, before allowing the script to continue.

If a decimal is put in, the number is rounded up. Furthermore, when a non-positive number is input, the loop does not run, and if "Infinity" is input, then the block runs forever. For example:

Animation — rather than coding each costume change and delay individually, the Repeat () block (with the Next Costume block) can be used. This can reduce unnecessary scripting.

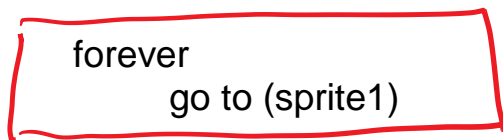
```
Repeat(8)
  wait(0.05)secs
  next costume
```

The **Forever** block is a Control block, a C block, and a Cap Block [**cap block** is a block that is designed to stop a block from being placed underneath it].

Used for infinite looping

Blocks held inside this block will be in a loop — just like the Repeat () block and the Repeat Until () block, except that the loop never ends (unless the stop sign is clicked, the Stop All block is activated, or the stop script block is activated within the loop). Due to this infinite loop, the block has no bump at the bottom; having a bump would be pointless, as the blocks below it would never be activated. For example:

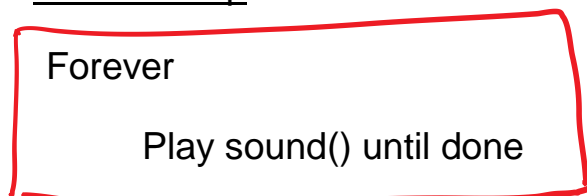
Keeping a sprite at another's location:



```
forever
  go to (sprite1)
```

A Scratch code block with a red border. It contains the text "forever" on the first line and "go to (sprite1)" on the second line.

A music loop



```
Forever
  Play sound() until done
```

A Scratch code block with a red border. It contains the text "Forever" on the first line and "Play sound() until done" on the second line.

### 15) What is a backdrop? Explain the process of changing a backdrop.

A **backdrop** is an image that can be shown on the Stage.

It is similar to a costume, except that it is shown on the stage instead.

They are in the backdrops library.

The Stage can change its look to any of its backdrops using the Switch Backdrop to () block. They can be named, edited, created, and deleted in the Paint Editor.

They are found in the backdrops tab in the Stage. New backdrops can be chosen from the library, painted, uploaded, or chosen as a surprise.

#### **Process of changing backdrops:**

There are 5 blocks related to backdrops:

In Looks:

- Switch Backdrop to ()
- Switch Backdrop to () and Wait
- Next Backdrop
- Backdrop ()

In Events:

- When Backdrop Switches to ()

For example:

```
If touching color _ then
  swtich backdrop to ()
  wait 1
```

Because of its simplicity, this block does not have a specific list of main uses; it is simply used to change the background of the Stage.

It can be used in animations, games, simulations—anything that requires changes between backgrounds.

## 16) Explain Costumes in Scratch

**COSTUME: alternate appearances of a sprite.**

Sprites can change their look to any of its costumes. They can be named, edited, created, and deleted, but every sprite must have at least one costume. One of the most common uses of costumes is to make an animation for a game or other project.

In looks :

Switch costume to ( )

Next costume

There are four main ways of getting a costume or background for a sprite or stage.

1. Drawing one using the inbuilt Scratch Paint Editor, by clicking on the costume or background tab, and clicking the "paint" button
2. Getting an image or multiple images that exist as files on the computer, either by clicking the "import" button under the costumes tab, or dragging in an image or images from an open folder.
3. Taking an image from a webcam, by clicking the "camera" button
4. Right clicking on a sprite and selecting "grab screen region for new costume" which enables you to define an area of the Scratch client to become a new costume (this can not be done for the stage, as this will create a new sprite instead)

By:

Dr.S.Anuradha