



CL1002

Programming Fundamentals

Lab 02

**Introduction to User Input and
Output, Variables, Operators and
Problem-Solving using Decision
and Iterative Structures using
Scratch Programming**

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

Fall 2025

AIMS AND OBJECTIVES

The aim of this lab is to equip students with scratch programming platform and apply fundamental programming concepts to demonstrate their skills. The objectives of this lab are as follows.

1. Teach students how to receive input from users and display output using Scratch.
 2. Teach students the concept of variables in Scratch and how to create, assign and use them within programs.
 3. Guide students in using Scratch's Operator blocks to perform arithmetic operations such as addition, subtraction, multiplication, and division.
 4. Introduce students to the relational (comparison) operator blocks in Scratch and show how they enable decision-making by comparing values
 5. Introduce students to the decision and iterative blocks in scratch programming to determine and repeat some series of statements respectively.
-

INTRODUCTION

In this lab we will be covering the following topics

1. User Input and Output.
2. Create and Assign Variables.
3. Arithmetic Operators
4. Relational Operators
5. Introduction to Decision and Iterative Structures

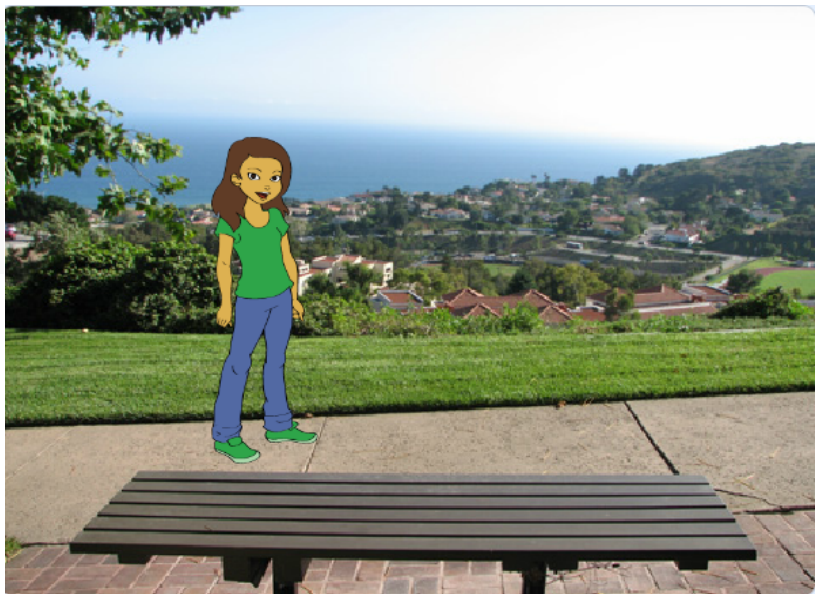
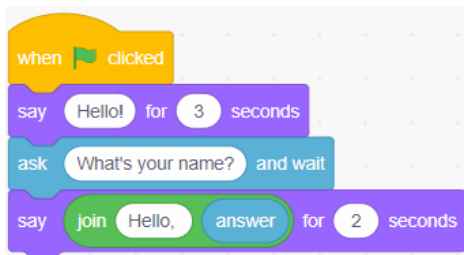
OBJECTIVE: 1 USER INPUT AND OUTPUT

• **Definition:** User input is a mechanism for receiving information from the user like asking a question and output is a mechanism of communicating information back to the user like displaying messages or sprites speaking

• **Importance:** It's fundamental to know how exactly you are going to communicate to and from the sprites.

EXAMPLE

Here we have a scenario where a sprite named as abby is walking on the track and asking what's your name and then we provide a name in the user input after that abby says Hello, and whatever the name you have provided.



Some more problems to solve

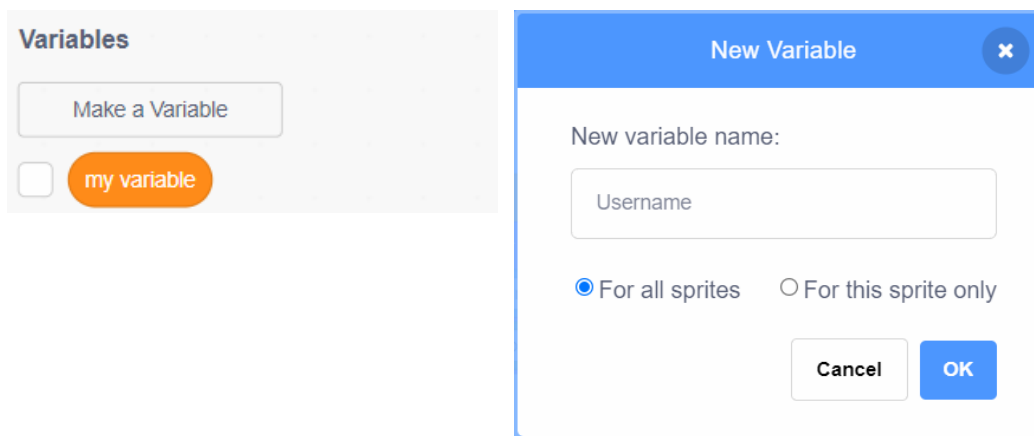
Problem#01—

Create a sprite that greets you by asking your name and student id.

OBJECTIVE: 2 CREATE AND ASSIGN VARIABLES

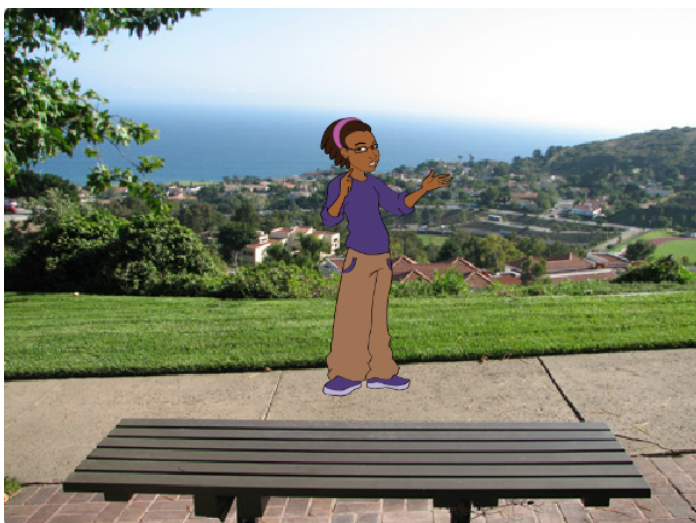
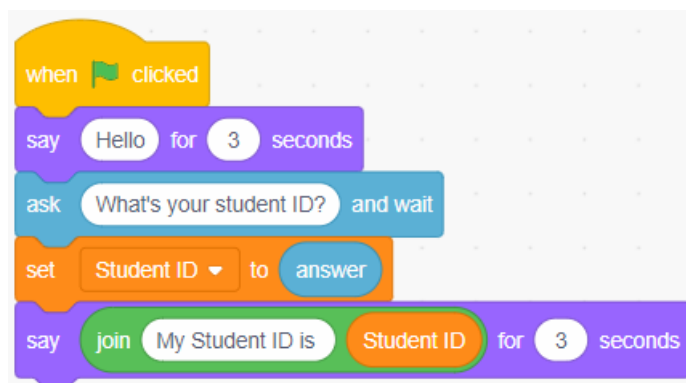
Variables are containers that hold values which can change over time. In Scratch, variables can store data such as numbers, text (strings), and Boolean values (True/False).

Students can create a variable using the “Make a variable” button in the Variables category. Each variable must have a name (e.g., “score”, “Username”) and a scope (for all sprites or just one sprite).



EXAMPLE

Here we have a scenario where a sprite named as avery is walking on the track and asking what's your student ID and then we provided the student ID and stored it into variable named as student ID and displayed the message with student ID.



Some more problems to solve—

Problem#02

Imagine you are designing student details record system. Create a scratch program that collects information from the user about a student and calculates the obtained marks and percentage based on the marks entered. The program should include the following steps:

- ✓ Ask the user to enter the student's ID, student's Name, marks for Subject 1, marks for Subject 2, and marks for Subject 3.
- ✓ Calculate the obtained marks by adding the marks for all three subjects.
- ✓ Calculate the percentage obtained by dividing the obtained marks by the maximum possible marks (assuming each subject has a maximum of 100 marks) and multiplying by 100.
- ✓ Display the student's details (ID and Name), the obtained marks, and the calculated percentage.

OBJECTIVE: 3 ARITHMETIC OPERATORS

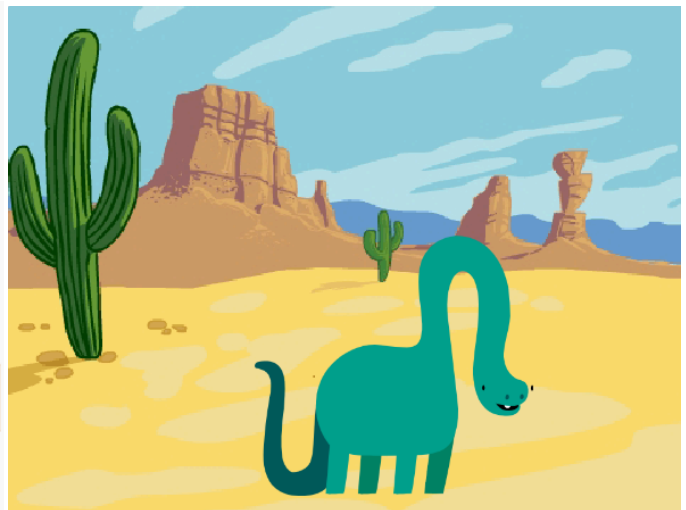
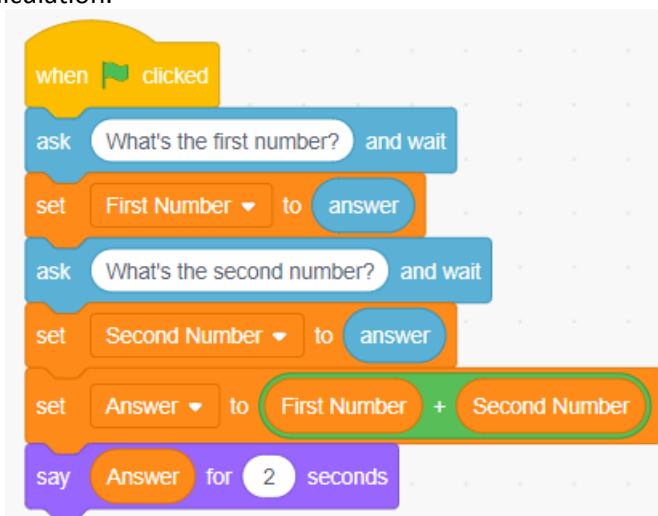
Operator Blocks: Scratch includes a suite of operator blocks (found under the *Operators* category) that handle arithmetic operations:

- Addition: +
- Subtraction: −
- Multiplication: ×
- Division: ÷

These blocks accept numerical inputs and return the result of the calculation

EXAMPLE

In the following scratch programming example, we are taking two user inputs what is the first number and what is the second number and storing the numbers in first number and second number variable respectively and besides that creating an answer which will store the result of the addition operation and will display the result of the calculation.



Some more problems to solve—

Problem#03

Create a scratch program that calculates the area of a rectangle. Ask the user for the length and width, then display the calculated area?

Problem#04

Create a scratch program that converts Celsius to Fahrenheit. Ask the user for a temperature in Celsius and display its equivalent in Fahrenheit using the formula: $\text{Fahrenheit} = (\text{Celsius} * 9/5) + 32$.

OBJECTIVE: 4 RELATIONAL OPERATORS

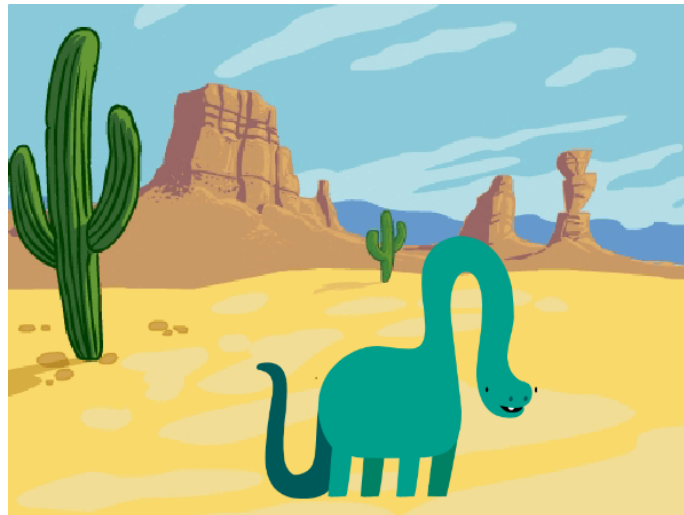
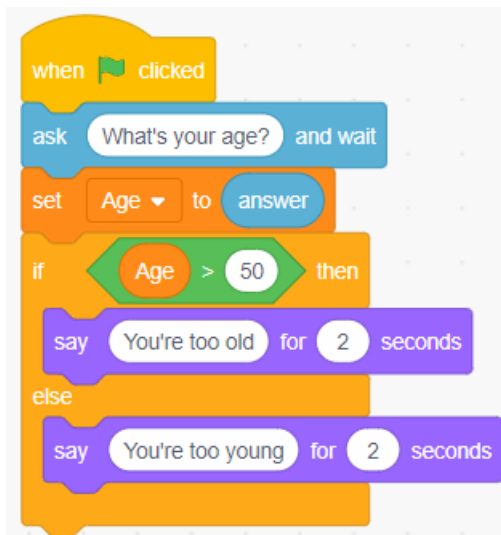
Relational Operators in Scratch are found in the Operators category and include:

- = (equal to)
- < (less than)
- > (greater than)
- ≤ (less than or equal to)
- ≥ (greater than or equal to)
- ≠ (not equal to)

These operators return Boolean results: either true or false.

EXAMPLE

In the following example we are asking the dinosaur its age and dinosaur provides his age and then using greater than relational operator we are determining whether the dinosaur's age is greater than 50 or not.



Some more problems to solve—

Problem#05

Create a scratch program to store a predefined password in a variable. Prompt the user, then check using relational operators whether the access is “granted” or “denied”.

OBJECTIVE: 5 DECISION AND ITERATIVE STRUCTURE

Decision Structure: A statement or a set of statements that is executed when a particular condition is “True” and ignored when the condition is “False”.

In scratch, we use the following control diagrams for decision structure.

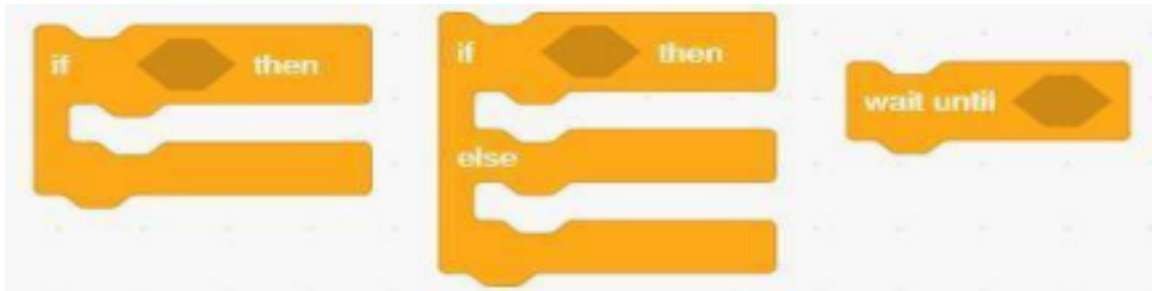


Figure: 5(a) If, If-else and wait until block

EXAMPLE

Given a number as an input by a user, check if the number is a negative number or a positive number.

Flowchart of Decision Structure	Scratch Diagram	Output

Figure: 5(b) Positive and Negative Number

Iterative Structure: The statements that cause a set of statements to be executed repeatedly either for a specific number of times or until some condition is satisfied are known as iteration statements.

In scratch, we use the following control diagrams for iterative structures:



Figure: 5(c) Repeat, Forever and Repeat Until

EXAMPLE

Set a counter to 1 and repeat until the given condition is satisfied. In this case, the given condition is counter=10.

Flowchart of Iterative Structure	Scratch Diagram	Output

Figure: 5(d) Counter Variable

EXAMPLE

Draw a pentagon with the help of repeat and pen diagrams. Repeat the shape for five times.


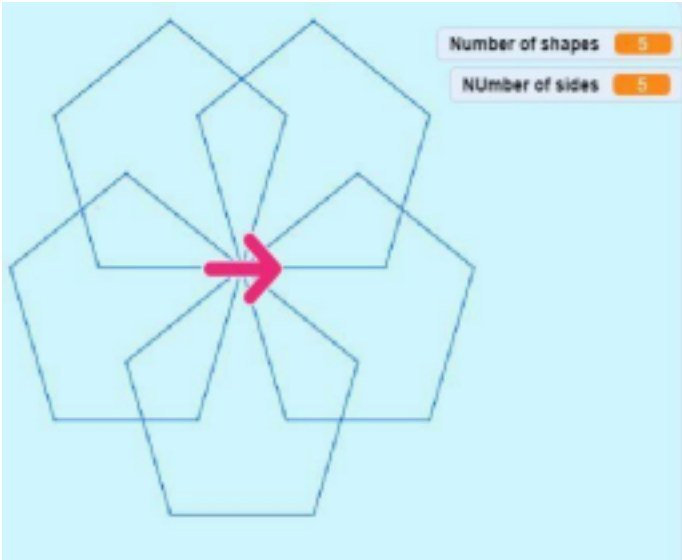
Scratch Diagram	Output
 <pre> when green flag clicked set Number of shapes to 5 set Number of sides to 5 pen down erase all repeat (Number of shapes) repeat (Number of sides) move 100 steps turn 360 / Number of sides degrees turn 360 / Number of shapes degrees change pen brightness by 50 </pre>	

Figure: 5(e) Pentagon