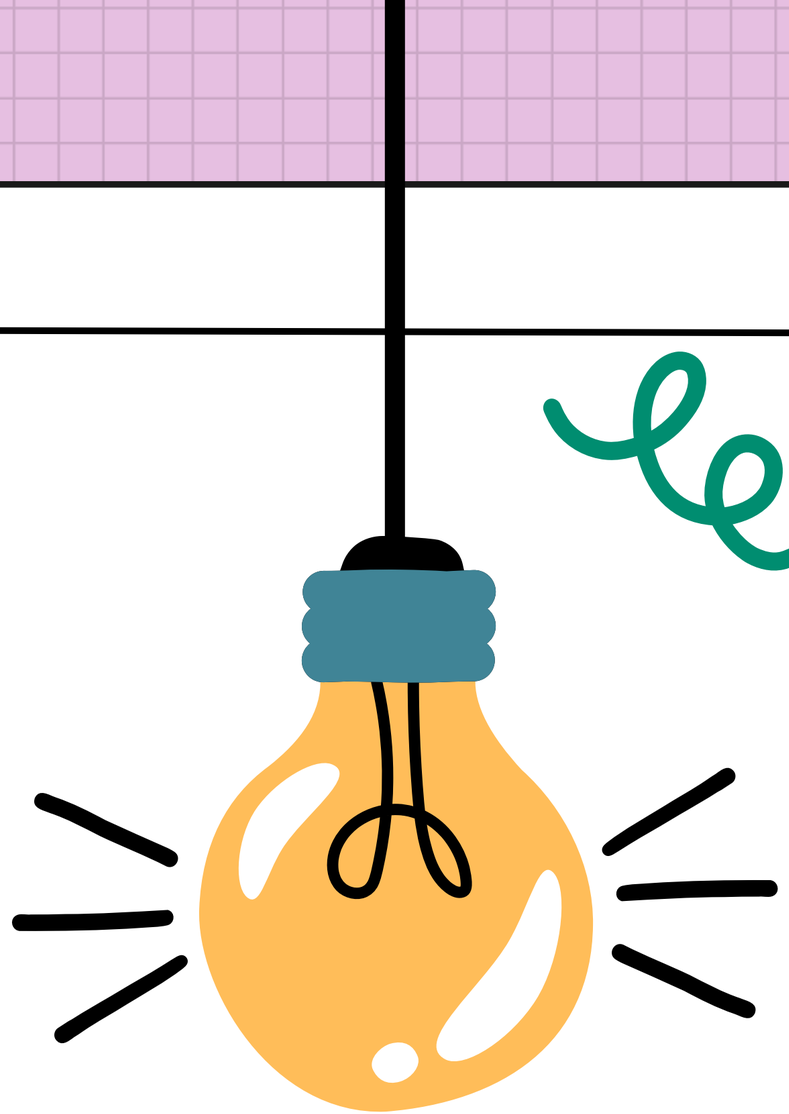
One Step In Changing Education Chain...

RED & WHITE®

Multimedia Education

C LANGUAGE SESSION



By Bhavya Nagar

Swipe

CONTENTS

Swipe

CONTENTS

Swipe

CONTENTS

Swipe

ARRAYS

Arrays are like big boxes 📦 where you can store many items of the same type together. You can think of them as rows of shelves in a store 🏪, each holding one item like a number or word. Accessing items in an array is like finding something on a specific shelf using an index 🎯. You can add or change items in an array, just like putting new items on a shelf or replacing them ➕🔄. Arrays help organize similar items in one place, making it easier to work with them in your program 🤔.

INTRODUCTION

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non orci hendrerit augue interdum lacinia at egestas dolor. Vivamus elementum pulvinar tempus.

INTRODUCTION

C language is all about communicating with the computer to get it to do specific tasks by giving it instructions to follow. These instructions are written in a way that the computer can understand and execute. It's like having a conversation with the computer, where you tell it what you want it to do, step by step, and it follows your instructions to accomplish the task. 🤖Whether it's performing calculations, displaying information on the screen, or handling data, C allows you to communicate your intentions to the computer effectively. 🚀📊🔧

VARIABLES

Variables in C are like small boxes 📦 where we can keep information. Just like you put your toys in a toy box or your clothes in a closet, variables store different kinds of data in a program to keep it safe.

TYPES OF INFORMATION

Variables can hold all sorts of information! They can store numbers 📊, words 📝, or even simple choices like "yes" or "no" ✅❌.

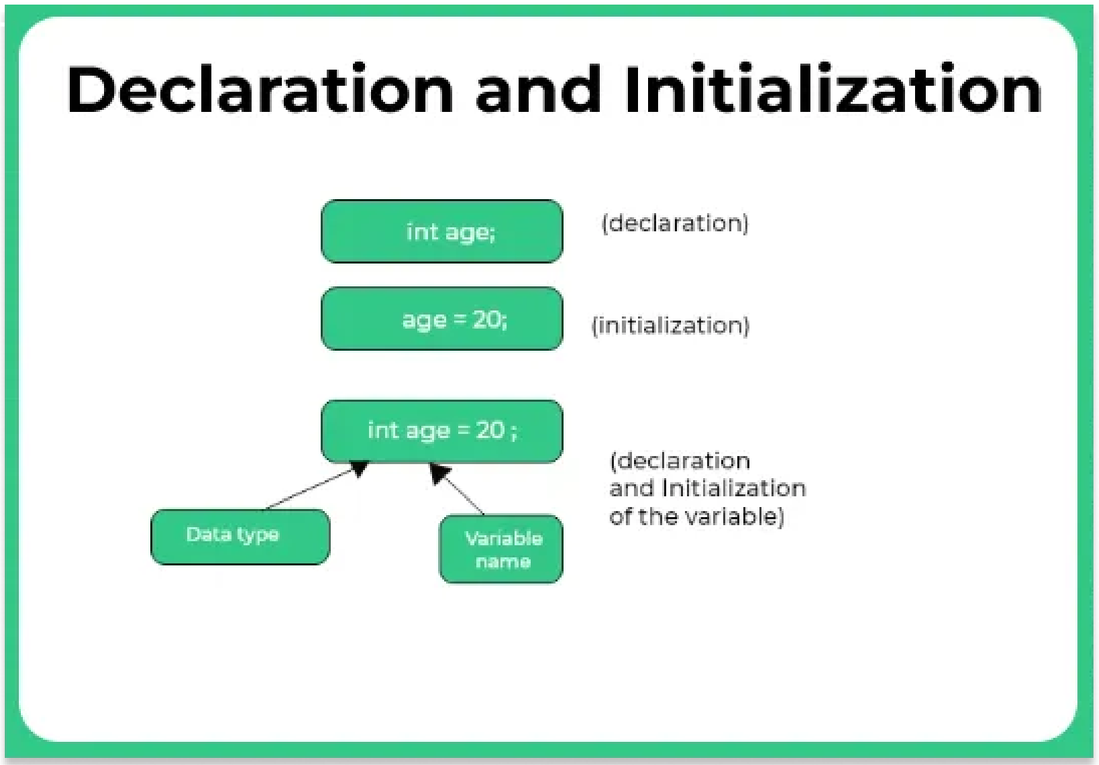
Swipe

VARIABLES

Main rules

1. Start with a letter or underscore: Variables must begin with a letter (a to z, A to Z) or an underscore (_). They can't start with a number or special characters. 🅰️🔡
2. Can contain letters, numbers, and underscores: After the first character, variables can have letters, numbers (0 to 9), or underscores (_). No spaces or special characters allowed! 🔢🔡
3. Be careful with reserved words: Variables can't have the same name as special words in C, called "reserved words" or "keywords." These are words like int, if, for, and so on. It's like trying to give your dog a name that's already taken by a famous movie star! 🚫🔑
4. Case sensitive: Variables in C are case-sensitive, meaning uppercase and lowercase letters are treated differently. So, age and Age would be two different variables. It's like telling your computer that "apple" and "Apple" are not the same thing! 🎩🍎
5. No special characters (except underscore): Variables can't contain special characters like $, %, @, etc., except for the underscore (_). It's like saying only letters and numbers are allowed to play in the variable club! 🚫$🚫

EXAMPLE



Swipe

DATATYPES

In C, data types are like containers 📦 that hold different kinds of information. There are integers 🔢 for whole numbers, floating-point numbers for numbers with decimal points, characters for single letters or symbols 🔤, and booleans for true or false values 🔵. These data types help us organize and manage different types of information in our programs, just like using different bags to store various items. 💡🛍️

integers (int)

These are whole numbers without any decimal points. They're like counting marbles 🌟, where you have a specific number of whole marbles.

```
int myVar;
```

floating-point numbers (float)

These are numbers with decimal points. They're like measuring water 💧 in a glass where you might have a fraction of the full glass.

```
float normalizationFactor = 22.442e2;
```

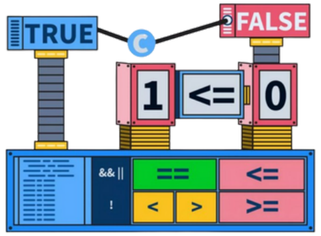
characters (char)

These are single letters or symbols. They're like writing a single letter 🅰 on a piece of paper.

```
char test = 'h';
```

booleans (bool)

These represent true or false values. They're like flipping a coin 🪙 to get either heads or tails.



Swipe

CONSTANT

Constants in C are like locked treasure chests 🔒 that hold information which cannot be changed during the program's execution. They're like secret codes that remain unchanged throughout the journey of your program.

TYPES OF CONSTANT

LET'S KNOW TYPES OF CONSTANT...

- What's Integer Constants?
- What's Floating-point Constants?
- What's Character Constants?
- What's String Constants?
- What’s Symbolic Constants?

These are whole numbers without decimal points, like 10 or -5. They're like counting stars in the sky 🌟.

These are numbers with decimal points, like 3.14 or -0.5. They're like measuring tiny drops of water 💧.

These are single characters enclosed in single quotes, like 'A' or '$'. They're like writing a single letter on a sticky note 📝.

These are sequences of characters enclosed in double quotes, like "Hello" or "C language". They’re like writing a message on a banner 🎉.

These are named values that cannot be changed during the program, like PI = 3.14. They're like giving a special name to a magic number 🎩.

Swipe

PRINTF

printf is a function in C language that allows you to display information on the screen. It's like having a magical printer 🖨️ that prints out messages for you to see.

SCANF

scanf is a function in C language that allows you to receive input from the user. It's like having a magical scanner 🔍 that scans the user's input and stores it in a variable for your program to use.

Swipe

FORMAT SPECIFIERS

1. Format specifiers in C, like %d for integers and %f for floats, are codes that tell the program how to format data. 📊🔤
2. They're like symbols in a secret code, representing different types of information such as numbers, characters, and strings. 🔍🔢🎯
3. Using the right specifier is important for correctly displaying or reading data, just like using the right symbol ensures the message is understood in a secret code! 🌟🔑

%d - Integers: This specifier is used for displaying or reading integer values. It's like a label for whole numbers! 🔢

%f - Floats: This specifier is used for displaying or reading floating-point values (numbers with decimal points). It's like a tag for numbers with fractions! 🎯

%c - Characters: This specifier is used for displaying or reading single characters. It's like a marker for individual letters or symbols! 🔤

%s - Strings: This specifier is used for displaying or reading strings (sequences of characters). It's like a code for writing sentences! 📜

lf - Double: This specifier is used for displaying or reading double-precision floating-point values. It's like a special marker for even more precise numbers! 🎯

Swipe

OPERATORS

Operators in programming are like tools 🔧 you use to perform different tasks on numbers, text, or other data. They're like magic spells 🪄 that make your program do cool things!

One Step In Changing Education Chain...
RED & WHITE®
Multimedia Education

TYPE OF OPERATOR

LET'S KNOW EACH TYPE OF OPERATOR...

- What's Assignment Operator (=)?
- What's Arithmetic Operators (+, -, *, /)?
- What's Comparison Operators (==, !=, <, >, <=, >=)?
- What's Logical Operators (&&, ||, !)?
- What's Increment (++) and Decrement (--) Operators
- What's Bitwise Operators (&, |, ^, ~, <<, >>)
- What's Conditional Operator (?:)
- What's Comma Operator (,)

This one's like giving a name 🏷️ to something. You use it to store a value in a variable.

These are your basic math friends! ➕➖✖️➗ They help you add, subtract, multiply, and divide numbers.

Comparison Operators ⚖️ help you compare values: equal to, not equal to, greater than, less than, greater than or equal to, or less than or equal to another value.

Logical Operators 🤔: && means "and", || means "or", and ! means "not". They help you make decisions based on conditions.

These are like adding or subtracting one from a number. It's like counting with a twist! 🔢

Manipulate bits (0s and 1s) to get the result you want.

Helps you make a decision between two options based on a condition.

Puts multiple expressions together in one line.

Swipe

DECISION MAKING

Decision-making is like choosing paths 🛤️ in a maze. They help your program decide what to do based on certain conditions

TYPES OF DECISION MAKING

LET'S KNOW EACH TYPE OF DECISION MAKING...

- What's If Statement?
- What's If-else Statement?
- What's Nested If-else Statement?
- What's Switch Statement?

It's like asking a question. If something is true, then do something

It's like giving two options. If something is true, do one thing; otherwise, do another.

It's like a series of questions inside questions. If one thing is true, you ask another question, and so on.

It's like choosing from a list of options. Based on a value, you select a specific option to execute.

Swipe

LOCAL & GLOBAL VARIABLES

Local variables 🏠 are like items in your house. They're declared inside a function and can only be used there. Once the function ends, they vanish, just like home items. 🏠

Global variables 🌍 are like neighborhood items. They're declared outside any function and usable from anywhere in your program. They remain in memory throughout, like neighborhood amenities. 🌍

LOOPS

A loop is like a merry-go-round 🎠 for your code! It repeats a block of instructions 🔄 until a condition is met, allowing you to automate repetitive tasks 🔄🔁🔂. It's like a loop-de-loop for your program's logic! 🔄📈

One Step In Changing Education Chain...
RED & WHITE®
Multimedia Education

TYPES OF LOOPS

LET'S KNOW EACH LOOPS...

- What's For Loop?
- What's While Loop?
- What's Do-While Loop?

A loop is like riding a roller coaster 🎢. You start at one point, do something, then loop back and do it again until you've done it enough times. 🎢

A loop is like playing tag 🔁. You keep running around until a certain condition is met, like being tagged. As long as the condition is true, you keep looping and doing something. 🔁

A loop is like a promise 🔁🔁. You do something at least once, then decide if you want to do it again. If yes, you do it again. 🔁🔁

Swipe

ARRAYS

Arrays are like big boxes 📦 where you can store many items of the same type together. You can think of them as rows of shelves in a store 🏪, each holding one item like a number or word. Accessing items in an array is like finding something on a specific shelf using an index 🎯. You can add or change items in an array, just like putting new items on a shelf or replacing them ➕🔄. Arrays help organize similar items in one place, making it easier to work with them in your program 🤔.

STRINGS

A string is like a string of beads 📿, where characters such as letters, numbers, and symbols are strung together in a sequence. Each character in a string has a specific place and order, similar to beads on a necklace 🔄. Strings are commonly used to store text, like words, sentences, or paragraphs 💬, and can be manipulated by cutting, copying, or changing parts of them ✂️. They are essential in programming for tasks like printing messages, reading user input, or storing information such as names and addresses 🤔.

FUNCTION

A function is like a magical tool 🛠️ in programming that performs a specific task when you use it, similar to casting a spell 🪄 to make something happen. You give a function some input (like ingredients for a potion 🧪) and it gives you back an output (like a magical potion 🧙). Functions are like little helpers 🧑‍🤝‍🧑 that you can call whenever you need them, and you can even create your own custom functions with your own special spells 🪄. They help organize your code and make it easier to understand, acting as magical helpers 🧞 that take care of repetitive tasks for you!

THANK YOU