Programming Fundamental

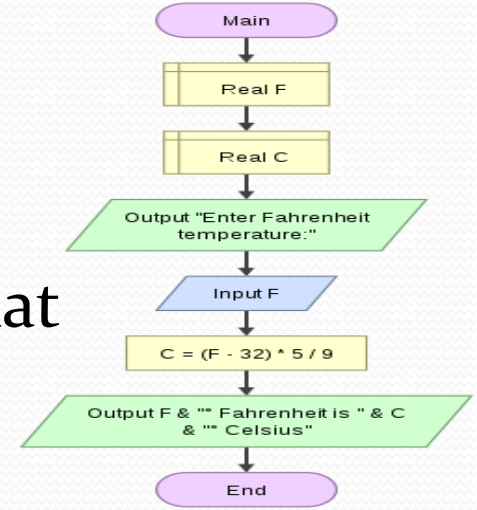
A Review

F2019105066
Aaira Ahmad

Programming Language Constructs

- **Variables:**

A **variable** is a known or unknown value that has been given a symbolic name. This allows the name to be used independently of the value. It is advisable that a meaningful name for readability and convenience. This name is known as the identifier.



Uses Of Variables

- **Variables** can represent numeric values, characters, character strings, or memory addresses.
  Variables play an important role in computer **programming** because they enable **programmers** to write flexible **programs**. Rather than entering data directly into a **program**, a **programmer** can **use variables** to represent the data.

Expressions

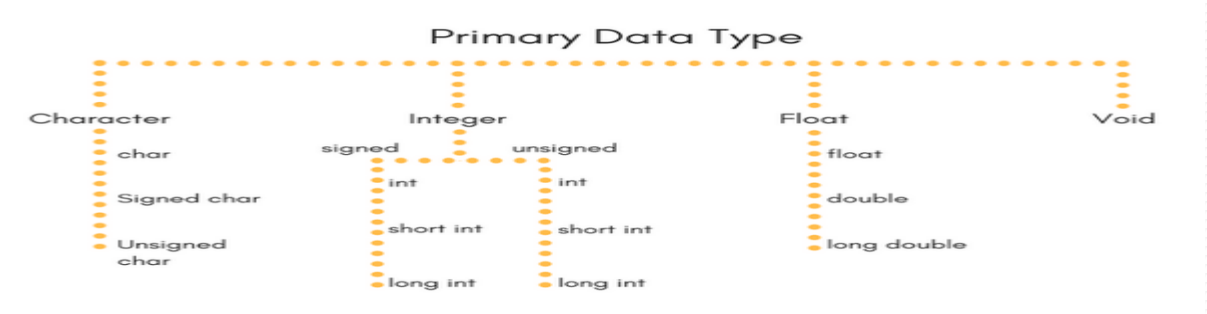
- An **expression** in a **programming** language is a combination of one or more constants, variables, operators, and functions that the **programming** language interprets (according to its particular rules of precedence and of association) and computes to produce another value.

Uses Of Expressions

- x and 5 are operands, and + is an operator.
  Expressions are **used** in **programming** languages, database systems, and spreadsheet **applications**. For example, in database systems, you **use expressions** to specify which information you want to see. These types of **expressions** are called queries.

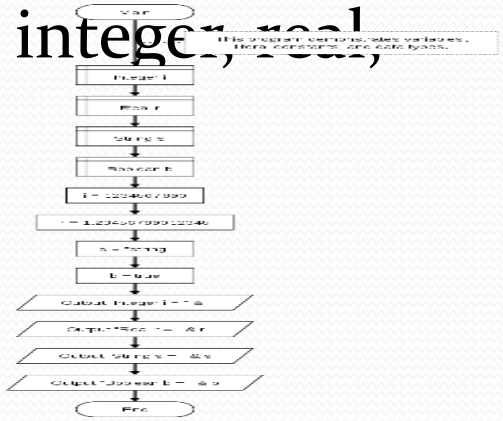
Data Types

- character, integer, float, and void are **fundamental data types**. Pointers, arrays, structures and unions are derived **data types**. char, Signed char, Unsigned char. Pointers are used for storing address of variables.



Uses Of Data Types

- A **data type** is a classification of **data** which tells the compiler or interpreter how the **programmer** intends to **use** the **data**. Most **programming** languages support various **types** of **data**, including integer, real, character or string, and Boolean.



Statements

- In computer **programming**, a **statement** is a syntactic unit of an imperative **programming** language that expresses some action to be carried out. A **program** written in such a language is formed by a sequence of one or more **statements**. A **statement** may have internal components (e.g., expressions).

C++ Includes Following Type Of Statements

- 1) expression statements;
- 2) compound statements;
- 3) selection statements;
- 4) iteration statements;
- 5) jump statements;
- 6) declaration statements;
- 7) try blocks;
- 8) atomic and synchronized blocks (TM TS).

Literal

- **literal**. In **programming**, a value written exactly as it's meant to be interpreted. In contrast, a **variable** is a name that can represent different values during the execution of the program. And a constant is a name that represents the same value throughout a program.

Types & Uses Of Literals

- Integer **Literals**: These are **used** to represent and store the integer values. ...
- Floating-Point **Literals**: These are **used** to represent and store real numbers. ...
- Character **Literal**: This refers to the **literals** that are **used** to store a single character within a single quote.

Operators

- **Operators** are symbols that tell the compiler to perform specific mathematical or logical manipulations. In this tutorial , we will try to cover the most commonly used **operators in programming**. Checks if the values of two operands are equal or not, if yes then condition becomes true.

Types Of Operators

- **Arithmetic Operators**. It includes basic **arithmetic operations** like addition, subtraction, multiplication, division, modulus **operations**, increment, and decrement.
- **Relational Operators**.
- Logical **Operators**.
- Assignment **Operators**.
- **Bitwise Operators**.

Thank you!