**Lecture 2**

**2.1 FLOWCHART**

The flowchart is a diagram which visually presents the flow of data through processing systems. This means by seeing a flow chart one can know the operations performed and the sequence of these operations in a system. Algorithms are nothing but sequence of steps for solving problems. So a flow chart can be used for representing an algorithm**.**

**2.1.1 Flowchart Symbols**

The basic symbols commonly used in flowchart drawing in Programs are: Process, input/output, Decision, Connector and Flow Lines, described as follows:

|  |  |
| --- | --- |
| **Symbol** | **Function** |
|  | starting or ending of the program |
|  | Indicates any type of internal operation inside the Processor or Memory |
|  | Used for any Input / Output (I/O) operation. Indicates that the computer is to obtain data or output results. |
|  | Used to ask a question that can  be answered in a binary format (Yes/No, True/False) |
|  | Used for connection, |
|  | Shows direction of flow. |

**Example:** draw a flowchart to Find the area of a circle of radius r.

****

**Example:** **Draw a**  **flowchart to find the greater number between two numbers .**

****

**Lecture 3**

**3.1 Character Set**

**Character set** : is a combination of English language (Alphabets) and math's symbols (Digits and Special symbols such that these characters are combined to form sentences and sentences are combined to form paragraphs. The character set are set of words, digits, symbols and operators that are valid in C++. The main types of Character Set are :-

**Alphabets** : are represented by A-Z or a-z. C++ Language is case sensitive programming language, so it takes different meaning for small and upper case letters. For example; Manpower and manpower are two different identifiers in C++. There are total 26 letters used in C++ programming.

**Digits** : Digits are represented by 0-----9 or by combination of these digits. By using the digits numeric constant can be written easily. There are total 10 digits used in the C++programming.

**Special Character:** we have a set of special characters that can be used in C++. All these characters are used for various purposes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **, (comma)** | **" (double conations)** | **. (dot)** | **: (colon)** | **; (semicolon)** |
| **?** | **'** | **!** | **|** | **\** |
| **/** | **~** | **\_** | **$** | **%** |
| **#** | **&** | **^** | **\*** | **-** |
| **+** | **<** | **>** | **(** | **)** |
| **{** | **}** | **[** | **]** | **=** |

**3.2 Identifier**

A C++ identifier is a name used to identify a variable, function, class, ). Functions, array, structures module, or any other user-defined item. An identifier starts with a letter A to Z or a to z or an underscore (\_) followed by zero or more letters, underscores, and digits (0 to 9).

**Examples**: Here are some examples of acceptable identifiers:

Mohd zara abc move\_name a\_123 cont1 flg\_min A30m yname50 \_temp j a23b9 retVal retval

**Examples** of invalid identifiers: 3v1, my name, True .

**Rules of naming Identifiers :**

* Identifiers can have alphabets, digits and underscore sign characters.
* They must not be a keyword or Boolean literal or null literal.
* They must not begin with a digit.
* They can be within a length (up to 127charachter).
* They cannot contain a space
* C++ is case sensitive i.e., upper-case letters and lower-case letters are treated differently.

**3.4 Keywords in C++**

Keywords are those words who has special meaning for compiler. We can't use keywords as variable name. The keywords should be in lower case letter. The keywords are also identifiers but cannot be user defined.

C++ has 32 Keywords as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Keywords** | | | |
| auto | double | int | struct |
| break | Else | long | switch |
| case | enum | register | typedef |
| char | extern | return | union |
| const | float | short | unsigned |
| continue | For | signed | void |
| default | Goto | sizeof | volatile |
| do | If | static | while |

**3.5 Data Types in C++**

Data types are used to define a variable. Data types represent the type of information present in a variable. Data types are the keywords, which are used for assigning a type to a variable.

**There are Different Data Types available in C++:**

1. **Integer Type :** Integer data type are like whole numbers, they also include negative numbers but does not support decimal numbers.

|  |  |  |
| --- | --- | --- |
| **Type** | **Storage size** | **Value range** |
| int | 2 or 4 bytes | -32,768 to 32,767 or  -2,147,483,648 to 2,147,483,647 |
| unsigned int | 2 or 4 bytes | 0 to 65,535 or 0 to 4,294,967,295 |
| short | 2 bytes | -32,768 to 32,767 |
| unsigned short | 2 bytes | 0 to 65,535 |
| long | 4 bytes | -2,147,483,648 to 2,147,483,647 |
| unsigned long | 4 bytes | 0 to 4,294,967,295 |

1. **Float-point Type :**Float data type allows user to store decimal values in a variable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Storage size** | **Value range** | **Precision** |
| float | 4 byte | 1.2E-38 to 3.4E+38 | 6 decimal places |
| double | 8 byte | 2.3E-308 to 1.7E+308 | 15 decimal places |
| long double | 10 byte | 3.4E-4932 to 1.1E+4932 | 19 decimal places |

1. **Character Type :** Character data type is used to store only one letter, digit, symbol at a time.

|  |  |  |
| --- | --- | --- |
| **Type** | **Storage size** | **Value range** |
| char | 1 byte | -128 to 127 or 0 to 255 |
| unsigned char | 1 byte | 0 to 255 |
| signed char | 1 byte | -128 to 127 |

**3.6 Variables in C++**

Variables are used to store values. variable name is the name of memory location where value is stored. It must be alphanumeric, only underscore is allowed in a variable name. It is composed of letters, digits and only underscore. It must begin with alphabet or underscore. It cannot be begin with numeric.

**Declaration of Variable**

Declaration will allocate memory for specified variable with garbage value.

**Syntax :**

Data-Type Variable-name;

**Examples :**

int a;

float b;

char c;

**Initialization of Variable**

Initialization means assigning value to declared variable. Every value will overwrite the previous value.

**Examples :**

a = 10;

b = 4.5;

c = 'a';

Character value must be enclosed with single quotes.

A=4.5;

if we assign decimal value to integer variable, it will accept only integer portion of value. In the above example variable a will accept 4 only.