Sher
Khan
Baloch

Fundamentals of Programming Notes

Fundamentals of Programming Notes

BSCS (1)

Sher Khan Baloch

What is Computer?

Computer is an electronic devices which Receive input, Process it, Store it, and Produce output according to given instructions.

What is Program?

Program is the set of instruction used to perform any specific task is called program.

What is Programming?

Programming is a technique, used to write different programs or it the source of communication with machine.

There are two types of programming languages

1. **Low Level Languages**
   Such programming languages which are near to machine and easy to understand for hardware but it is difficult to write for Humans/Programmers.
   a. **Machine Language or Binary Language**
      This deals only with the 0 and 1. It is difficult to write program, understand program, debug program and maintain program.
   b. **Assembly Language**
      It deals with the Mnemonics or symbolic codes like Move, Add, and Sum etc. It is one step high to Machine Language but still hard to code.
2. **High Level Languages**
   Such programming languages which is easy to Program for human but it is difficult to understand for the machine because machine only understand 0 and 1 and these languages are like English language like if, else, switch, printf, echo, cout, cin. These languages are C. C++, C#, Java, Python etc. Then here we need language Processors.

Language Processors:

Processors are the translators used to translate Source Code into machine understandable code or Object Code.
There are three types of Language Processors

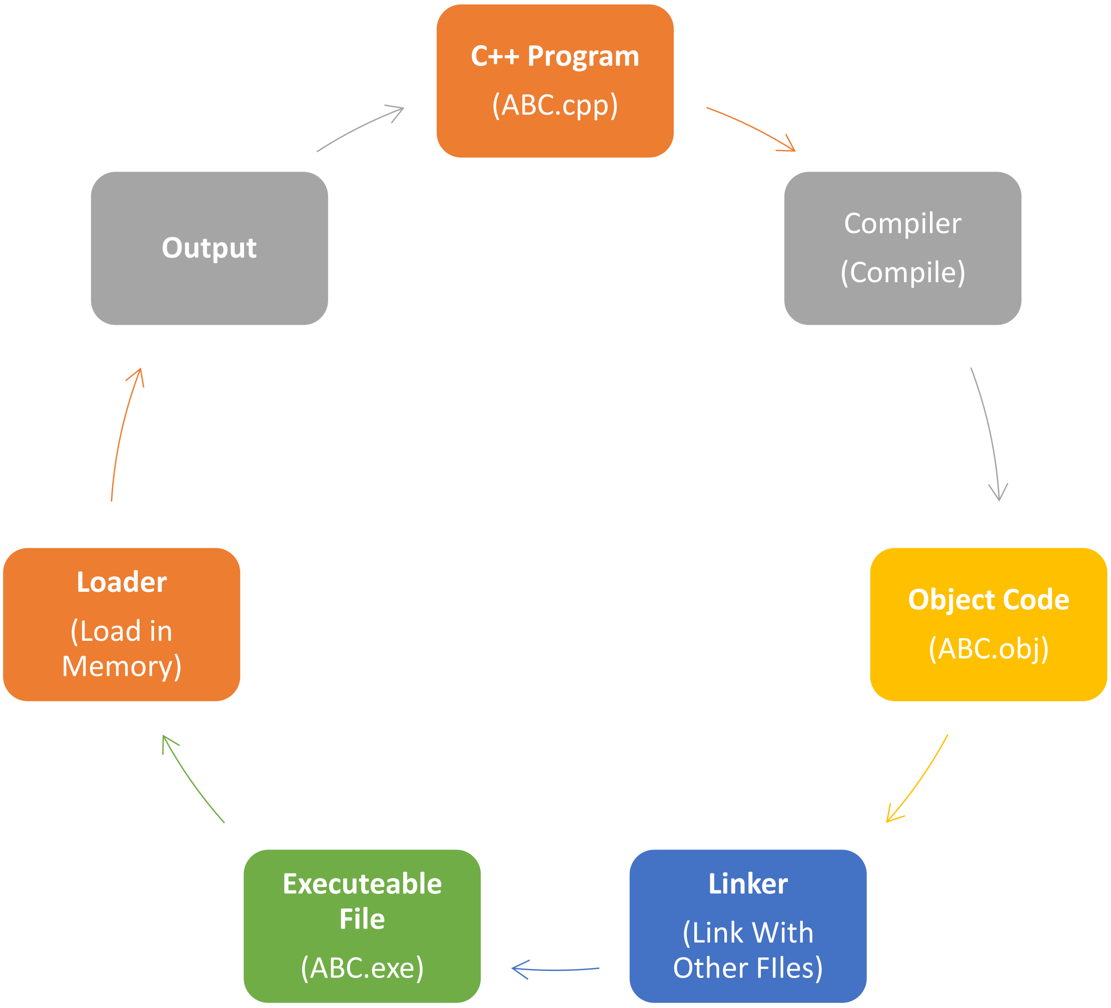
1. **Assembler**
   It is a translator that translate the Assembly language code into machine understandable code.
2. **Interpreter**
   It is a translator that translate the High Level language code into machine understandable code. It compile and execute one instruction at a time.
3. **Compiler**
   It is a translator that translate the High Level language code into machine understandable code. It compile and execute whole program at a time.

Errors

Errors are the mistake or bugs in a program. There can be three types of Errors.

1. **Syntax Error**
   Syntax is the rule and regulations of a program. This error can be spelling mistake, undefined words, and incorrect punctuations. Syntax error occurs when our program is compiling it is also known as Compile time error.
2. **Run Time Error**
   Such mistakes which occurs during the exception of a program.
   Example: int a = 100/0;
3. **Logical Error**
   Such errors which produce output but that output is not desired or correct output.
   Example: If we want to plus two values but it is subtracting these values so it is a logical error.

Structure of C++ Program



IDE and Structured of C++

IDE: Integrated Development Environment
IDE contains two parts

1. **Editor**
   Where we write the syntax of program.
2. **Console**
   Where output of program is generated.

Libraries and Functions

1. **Libraries**
   Libraries contains information about functions, reserve words. It contains files or namespace also known as header files. The extension of the header file is .h. We can add namespace directly.

   ```
   #include <iostream> // Header Files
   using namespace std; // Namespace
   ```

2. **Main Function**
   Main functions contains body { } and statement terminator or semicolon (;).

   ```
   int main()
   {
   	cout << “Sher Khan Baloch”;
   }
   ```

Pre-Processor Directives

Such directives which starts with # symbol such instruction which is already processed or the computer invoke them before compiling the program.

There are two types of Pre-Processor Directives.

1. **Include**
   Adding different files or namespaces.

   ```
   #include <File Name>
   #include <iostream> // Namespaces
   #include <stdio.h> // Header Files
   #include “myFile.h” // Custom File Created By Us.
   ```

2. **Define**
   It is used for making macros.

   ```
   #define size 5;
   ```

Keywords and Reserve Words:

1. **Keywords**
   Keywords are the pre-defined words in our language that has some specific meaning in that line. Some words will be used for data types, conditions, and loops etc.
   Example: int, float, char, if, else, switch, case, break, for, while, do, sizeof etc.
2. **Reserved Word**
   Such words which are not the part of a language but part of header files or namespaces.
   Example: String (std), endl (iostream) and Null (\0) etc.

- Keywords cannot be redefined.
- Reserved Words can be redefined. We can create our own reserved words or can change the existing reserve words.
- Reserved words needs libraries or namespaces but keywords does not required these things.

Output Functions

1. **Printf**
   Full Name: Print Function.
   Header File: stdio.h
   Syntax: printf("Sher Khan Baloch");
2. **Cout**
   Full Name: Console Output.
   Header File: iostream
   Syntax: cout << "Sher Khan Baloch";
3. **Cprintf**
   Full Name: Color Print Function.
   Header File: conio.h
   Syntax:
   ```
   textbackground(WHITE);
   textcolor(RED);
   cprintf("Sher Khan Baloch");
   ```
4. **Put**

cout << "Sher Khan Baloch"; Here **<<** is a **Stream Insertion Operator.**

We can also write our own reserved word instead of cout.

```
Int main()
{
        ostream &SherKhan = cout;
        SherKhan << "Hello World";
}
```

Escape Sequence

It starts with \ and attach one character with it. It is used for the formatting of our output.

Examples:

1.	**\n**	→	New Line	→	cout << "Sher Khan Baloch \n";
2.	**\t**	→	Tab Space	→	cout << "Sher \t Khan \t Baloch";
3.	\	→	Back Slash	→	cout << "Hello World \\ Global";
4.	**\"**	→	Double Quote		
5.	**\b**	→	Backspace		
6.	**\r**	→	Carriage Return		

Format Specifiers

cout << 5+9 → 14

cout << "5+9" → 5+9

1.	%d	→	Integer Value
2.	%i	→	Integer Value
3.	%f	→	Float/Decimal Value
4.	%e	→	Float/Decimal Value
5.	%o	→	Octal
6.	%x	→	Hexa
7.	%c	→	Char
8.	%s	→	String

printf("5 + 9 = %d", 5+9);

%d means print the integer value here which we are calculating after it.

Output: 5 + 9 = 14

printf("Value of Pie = %f", 3.14);

Output: Value of Pie = 3.14

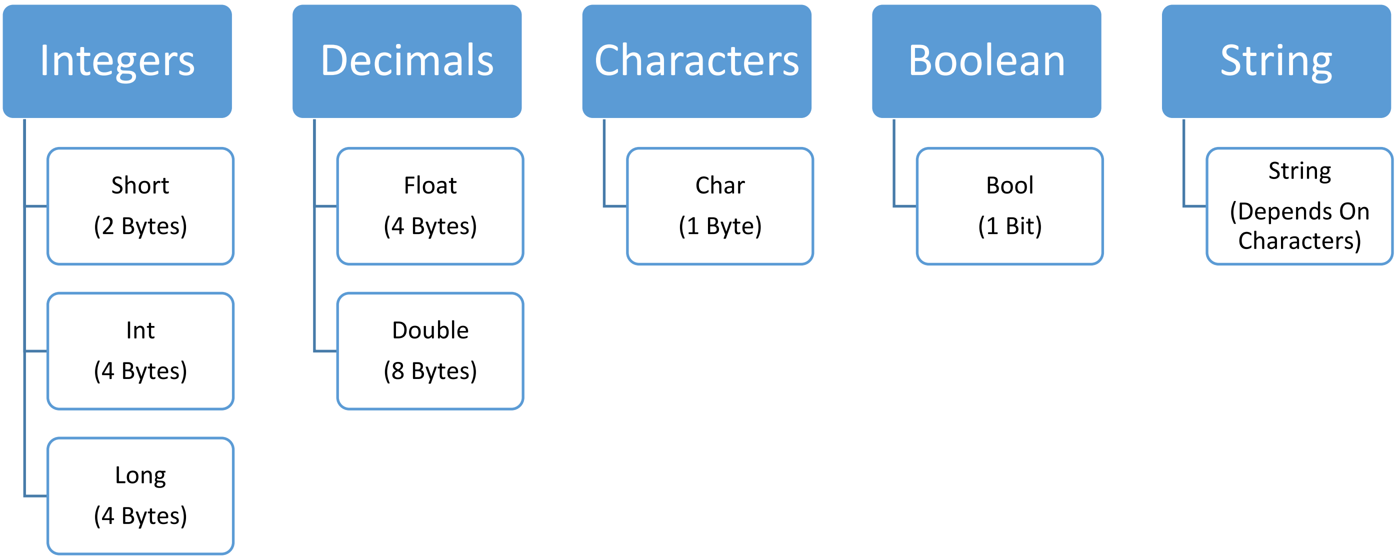
printf("Value = %2f", 5.456");

%2f means print only 2 decimals of number. It is also known as output width.

Variables and Data Types

Variables: Variables are the named memory locations which contains some value that can be changed.

Data Types: Data Types define the type of data stored in a variable.



Integers

Short (2 Bytes)

Int (4 Bytes)

Long (4 Bytes)

Decimals

Float (4 Bytes)

Double (8 Bytes)

Characters

Char (1 Byte)

Boolean

Bool (1 Bit)

String

String (Depends On Characters)

Declaration of Variable:

```
Datatype varibaleName;
int num;
```

Assign Value To Varibale:

```
variableName = Value;
num = 10;
```

Initialization of Variable:

```
Datatype variableName = Value;
int num = 10;
```

Other Examples:

```
int x = 10;
float x = 5.95;
char ch = ‘A’; or char ch = 65;
bool isActive = true;
string name = “Sher Khan Baloch”;
```

Rules for Naming the Variables:

1. Keywords cannot be variable name.
2. Blank spaces are not allowed.
3. Numbers, Alphabet, Underscore are allowed.

4. Operator, Punctuation, Special Symbols are not allowed.
5. We can’t use duplicate names.
6. Digits can’t be first letter of variable.
7. Maximum limit of variable name is 32 letters.

Operators

Expression = Operator and Operands

10 + 5, Here **10+5** is Expression and **10** and **5** are operands and **+** is operator.

Operator: Operators are the symbol used to perform any operation.

There are three types of operators:

1. Uniary
   An operator that perform operation on a single operand. e.g. !a, a++, &a
2. Binary = a < b, a = 10, a + b
3. Ternary = ?, :

Operands: Operands are the things or variable or a memory location to whom these operations are going to be perform.

There are three types of operands:

1. Variables or Memory Location
2. Registers
3. Imigate Value

1. **Arithmetic Operators (Binary Operator):**
   1. Addition (+)
   2. Subtraction (-)
   3. Multiplication (*)
   4. Division or Quotient (/)
   5. Modulus or Remainder (%)

2. **Relational Operators (Binary Operator):**
   These are used for Comparison between two operands. These produce Result in either true (1) or false (0).
   1. Less Than (<)
   2. Less Than Equal To (<=)
   3. Equal To (==)
   4. Not Equal To (!=)
   5. Greater Than (>)
   6. Greater Than Equal To (>=)

3. **Logical Operators**
   1. NOT (!) (Toggle The Value True to False and False to True)
   2. AND (&&) (Joint Operator and Use For Condition)
   3. OR (||) (Joint Operator and Use For Condition)

Truth Table:

A	B	!A	&&	||
True	True	False	True	True
True	False	False	False	True
False	True	True	False	True
False	False	True	False	False