**Fundamentals of Programming**

**FUNDAMENTALS IN PROGRAMMING**

* A computer program is a sequence of instructions written using a Computer Programming Language to perform a specified task by the computer.

1. Go straight
2. Drive half kilometer
3. Take RIGHT
4. Drive around one kilometer
5. Search for UCC at your LEFT side

**print "Hello, World!"**

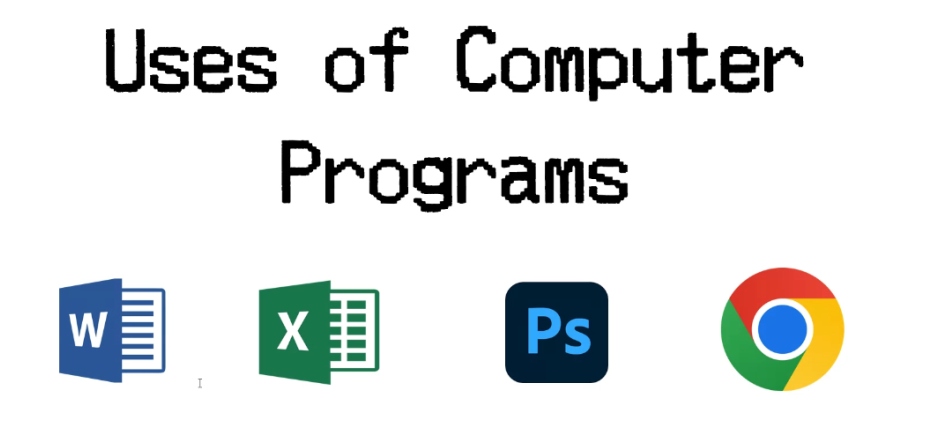
**COMPUTER SOFTWARE**

**PROGRAM CODING**

**COMPUTER PROGRAMMING**

* Java
* C
* C++
* Python
* PHP
* Perl
* Ruby

**Uses of Computer Programs**



**Computer Programmer**

* Someone who can write computer programs

**Algorithm**

* an **algorithm** is a step-by-step procedure to resolve any problem.

1. Get a list of numbers L1, L2, L3....LN
2. Assume L1 is the largest, Largest = L1
3. Take next number Li from the list and do the following
4. If Largest is less than Li
5. Largest = Li
6. If Li is last number from the list then
7. Print value stored in Largest and come out
8. Else repeat same process starting from step 3

* Programming Environment
* Basic Syntax
* Data Types
* Variables
* Keywords
* Basic Operators
* Decision Making
* Loops
* Numbers
* Characters
* Arrays
* Strings
* Functions
* File I/O

**Computer Programming - Environment**

* A working Internet connection to connect to the Internet
* A Web browser such as Internet Explorer, Chrome, Safari, etc.

**TEXT EDITOR**

**COMPILER**

**INTERPRETER**

**Text Editor**

• A text editor is a software that is used to write computer

programs.

**C++ Fundamentals in Programming**

**Bjarne Stroustrup**

*Buh jaar nay strau struhp*

1979 at Bell Labs

* more libraries
* more functions

**Why to Learn C++**

* C++ is very close to hardware
* C++ programming gives you a clear understanding about Object Oriented Programming.
* C++ is one of the every green programming languages and loved by millions of software developers.
* C++ is the most widely used programming languages in application and system programming.
* C++ really teaches you the difference between compiler, linker and loader, different data types, storage classes, variable types their scopes etc.

**Applications of C++ Programming**

* Application Software Development
* Programming Languages Development
* Computation Programming
* Games Development

**Object-Oriented Programming**

* Encapsulation
* Data hiding
* Inheritance
* Polymorphism

**C++ Basic Syntax**

* **Object** — Objects have states and behaviors.
* **Class** — A class can be defined as a template/blueprint that describes the behaviors/states that object of its type support.
* **Methods** — A method is basically a behavior.
* **Instance Variables** — Each object has its unique set of instance variables.

**C++ Program Structure**

#include <iostream>

using namespace std;

int main() {

std::cout << "Hello World!";

return 0;

}

**C++ Identifiers**

* mohd
* zara
* abc
* myname50
* \_temp
* j
* Move\_name
* a23b9
* a\_123
* retVal

**C++ Keywords**

* asm
* auto
* bool
* break
* case
* catch
* char
* class
* const
* const\_cast
* continue
* default
* delete
* do
* double
* dynamic\_cast
* else
* enum
* explicit
* export
* extern
* false
* float
* friend
* goto
* if
* inline
* long
* mutable
* namespace
* operator
* private
* protected
* public
* register
* reinterpret\_
* return
* short
* signed
* sizeof
* static
* static cast
* struct
* switch
* template
* cast
* this
* throw
* true
* try
* typedef
* typeid
* typename
* umon
* unsigned
* using
* virtual
* void
* volatile
* wchar\_t
* while

**Whitespace in C++**

• Statement 1

int age;

• Statement 2

fruit = apples + oranges; // Get the total fruit

**Comments in C++**

/\* This is a comment \*/

/\* C++ comments can also

\* span multiple lines

\*/

#include <iostream>

using namespace std;

main() {

cout << "Hello World"; // prints Hello World

return 0;

}

**C++ Data Types**

**Primitive Built-in Types**

|  |  |
| --- | --- |
| Type | Keyword |
| Boolean | bool |
| Character | char |
| Integer | int |
| Floating point | float |
| Double floating point | double |
| Valueless | void |
| Wide character | wchar\_t |

|  |  |  |
| --- | --- | --- |
| Type | Typical Bit Width | Typical Range |
| char | 1byte | -127 to 127 or 0 to 255 |
| unsigned char | 1byte | 0 to 255 |
| signed char | 1byte | -127 to 127 |
| int | 4bytes | -2147483648 to 2147483647 |
| unsigned int | 4bytes | 0 to 4294967295 |
| signed int | 4bytes | -2147483648 to 2147483647 |
| short int | 2bytes | -32768 to 32767 |
| unsigned short int | 2bytes | 0 to 65,535 |
| signed short int | 2bytes | -32768 to 32767 |
| long int | 8bytes | -2,147,483,648 to 2,147,483,647 |
| signed long int | 8bytes | same as long int |
| unsigned long int | 8bytes | 0 to 4,294,967,295 |
| long long int | 8bytes | -(2^63) to (2^63)-1 |
| unsigned long long int | 8bytes | 0 to 18,446,744,073,709,551,615 |
| float | 4bytes |  |
| double | 8bytes |  |
| long double | 12bytes |  |
| wchar\_t | 2 or 4 bytes | 1 wide character |

**C++ Variable Types**

**Type & Description**

**bool**

Stores either value true or false.

**char**

Typically a single octet (one byte). This is an integer type.

**int**

The most natural size of integer for the machine.

**float**

A single-precision floating point value.

**double**

A double-precision floating point value.

**void**

Represents the absence of type.

**wchar\_t**

A wide character type.