

# C++ Language Basics Summer School - 2010

## Lecture 1

Indian Institute of Technology, Kanpur

May 12, 2010

# 1 PROLOGUE

# Instructor

- Who? Shaktimaan Singh Sengar
- Where? 4<sup>th</sup> Year Undergraduates  
Department of Computer Science and Engineering  
IIT Kanpur

# Course Details

The objectives of this course are:

- Learning to solve problems **algorithmically**
  - Algorithm is an effective method for solving a problem expressed as a finite sequence of instructions
- Learning to convert programs into **algorithms**
- Learning a basic programming language **C++**

## Books and references

- C++: The Complete Reference, 4th Edition by Herbert Schildt
- The C++ Programming Language by Bjarne Stroustrup
- Online tutorials
  - <http://www.cplusplus.com/doc/tutorial>

# Basics

- Why Computers?

# Basics

- Why Computers?
- A computer can carry out any computational task.
- Computation is a process following a well-defined model.
  - E.g. computing the value of  $(a + b) * (c + d)$ .
  - E.g. computing the minimum of  $n$  numbers.
- It has been formally proved that given sufficient memory and time a computer can carry out any computational task.

Computers are dumb!



Computers are dumb!

- Why programming languages?

## Computers are dumb!

- Why programming languages?
  - We need a medium to interact with computers.
  - Need to write program in such a way that they can be understood by the computer.

## Computers are dumb!

- Why programming languages?
  - We need a medium to interact with computers.
  - Need to write program in such a way that they can be understood by the computer.
- Program is a sequence of instructions written to perform a specified task for a computer.
- Program's instructions are executed in a central processor.
- The program has an executable form which is executed by the CPU.
- The same program has a human-readable source code form.
- The human-readable source code when compiled gives the executable form.

# Binary Format

- In computer everything is stored in the binary format (a string of 0's and 1's)
- The computer has components who can only understand in the binary format.
- Binary representation of 7 is 0000111 and of 9 is 0001001.

# Programming Languages

The programming languages are broadly classified into two categories

- **Low Level Programming Languages:**
  - Small or nonexistent amount of abstraction
  - Close to the hardware.
  - **Examples:** Assembly Language, Machine Language.
- **High Level Programming Languages:**
  - Strong abstraction from the details of the computer
  - More user friendly.
  - Need for an compiler
  - **Examples:** C,C++,Java,Haskell.

# Compiler

- A computer program which translates **source code** to **target code**.
- Takes a High Level Language as input.
- Translates it into the Low Level Language which it gives as output.
- Compiles the whole code at a time and converts it into an executable object file (**.o file**).
- All the syntax and memory referencing errors are detected in the compilation phase.

# C++ Basics

- C++ programs are saved with extensions `.C`, `.cc`, `.cpp`, `.cxx` depending on the platform you are working on.
- Compile using `g++` compiler:  
`g++ first.cc -o first`
- Here `first.cc` is the name of the program file. `-o` converts it into the object file named `first.o`
- To run the program:  
Type `./first` at the command prompt.
- Those working on windows can use a developer's platform like `devC++`.

# A First Example

```
// My first program in C++. This acts as a comment.  
#include <iostream>  
using namespace std;  
int main ()  
{  
  cout << "Hello World!";  
  return 0;  
}
```

Output of the given program will be: [Hello World](#).



# A First Example

- Lines beginning with a hash sign (#) are directives for the preprocessor.
- Tells the program to include `iostream` standard file which includes the declaration of basic input output commands.

# A First Example

- Lines beginning with a hash sign (`#`) are directives for the preprocessor.
- Tells the program to include `iostream` standard file which includes the declaration of basic input output commands.
- `int main()` corresponds to the beginning of the definition of the main function.
- The main function is the point by where all C++ programs start their execution, independently of its location within the source code.

# A First Example

- The word `main` is followed in the code by a pair of parentheses `()` as this is a function declaration.
- The line `cout <<" Hello World!";` is a statement.
- `cout` is the name of the standard output stream in C++. The entire statement aims to insert a sequence of characters into this stream.
- Notice that the statement ends with a semicolon character `(;)` which marks the end of a statement in C++.

# A First Example

- The return statement causes the main function to finish which must be followed by a 0.
- A return code of 0 for the main function is generally interpreted as the program worked as expected without any errors during its execution.
- We could have written the entire main function in a single line:  

```
int main () { cout <<"Hello World!"; return 0; }
```

# Comments

Comments are parts of the source code disregarded by the compiler. They simply do nothing.

- //line comment
- /\*block comment\*/

```
/* My first program in C++  
with more comments */.
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main ()
```

```
{
```

```
cout <<"Hello World!"; // print hello world
```

```
return 0; // return nothing
```

```
}
```