

Tokens in C and C++

Talk to a Teacher

<http://spoken-tutorial.org>

National Mission on Education through ICT

<http://sakshat.ac.in>

Ashwini R Patil

IIT Bombay

4 April 2012



Learning Objectives



Learning Objectives

- ▶ How to define and use tokens



Learning Objectives

- ▶ **How to define and use tokens**
- ▶ **With the help of an example**



Learning Objectives

- ▶ **How to define and use tokens**
- ▶ **With the help of an example**
- ▶ **Some common errors and their solutions**



System Requirements



System Requirements

- ▶ **Ubuntu OS v. 11.10**

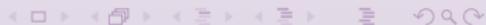


System Requirements

- ▶ **Ubuntu OS v. 11.10**
- ▶ **gcc and g++ Compiler v. 4.6.1**



Introduction



Introduction

- ▶ **Token is a generic word for Data types, Variables, Constants and Identifiers**



Keywords



Keywords

- ▶ **Keywords have fixed meanings that cannot be changed**



Keywords

- ▶ **Keywords have fixed meanings that cannot be changed**
- ▶ **Keywords cannot be used as variable names**



Keywords

- ▶ **Keywords have fixed meanings that cannot be changed**
- ▶ **Keywords cannot be used as variable names**
- ▶ **There are 32 keywords in C**

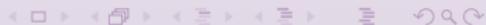


Keywords

- ▶ **Keywords have fixed meanings that cannot be changed**
- ▶ **Keywords cannot be used as variable names**
- ▶ **There are 32 keywords in C**
- ▶ **To name some auto, break, case, char, enum, extern, etc.**



Constants



Constants

- ▶ **Constants are fixed values**



Constants

- ▶ **Constants are fixed values**
- ▶ **They do not change during the execution of program**



Constants

- ▶ **Constants are fixed values**
- ▶ **They do not change during the execution of program**
- ▶ **There are two types of constants**



Constants

- ▶ Constants are fixed values
- ▶ They do not change during the execution of program
- ▶ There are two types of constants
 - ▶ Numeric constants



Constants

- ▶ **Constants are fixed values**
- ▶ **They do not change during the execution of program**
- ▶ **There are two types of constants**
 - ▶ **Numeric constants**
 - ▶ **Character constants**



Data Types

Type	Specification	Format specifier
------	---------------	------------------



Data Types

Type	Specification	Format specifier
Integer	int	%d



Data Types

Type	Specification	Format specifier
Integer	int	%d
Floating point	float	%f



Data Types

Type	Specification	Format specifier
Integer	int	%d
Floating point	float	%f
Character	char	%c



Data Types

Type	Specification	Format specifier
Integer	int	%d
Floating point	float	%f
Character	char	%c
Double	double	%lf



Range of Data Types

Data Type

Range of Values



Range of Data Types

Data Type	Range of Values
Integer	-32,768 to 32,767



Range of Data Types

Data Type	Range of Values
Integer	-32,768 to 32,767
Floating point	3.4E +/-38



Talk to a Teacher

Range of Data Types

Data Type	Range of Values
Integer	-32,768 to 32,767
Floating point	3.4E +/-38
Character	-128 to 127



Talk to a Teacher

Range of Data Types

Data Type	Range of Values
Integer	-32,768 to 32,767
Floating point	3.4E +/-38
Character	-128 to 127
Double	1.7E +/-308



Variables



Variables

- ▶ **Variable is a data name**



Variables

- ▶ **Variable is a data name**
- ▶ It may be used to store a data value



Variables

- ▶ **Variable is a data name**
- ▶ **It may be used to store a data value**
- ▶ **The values can change when a program runs**



Variables

- ▶ **Variable is a data name**
- ▶ **It may be used to store a data value**
- ▶ **The values can change when a program runs**
- ▶ **Before using a variable it must be declared**



Variables

- ▶ **Variable is a data name**
- ▶ **It may be used to store a data value**
- ▶ **The values can change when a program runs**
- ▶ **Before using a variable it must be declared**
- ▶ **We should try to give meaningful names to variables**



Talk to a Teacher

Variables

- ▶ **Variable is a data name**
- ▶ **It may be used to store a data value**
- ▶ **The values can change when a program runs**
- ▶ **Before using a variable it must be declared**
- ▶ **We should try to give meaningful names to variables**
- ▶ **eg. john, marks, sum, etc.**



Identifiers



Identifiers

- ▶ Identifiers are user defined names



Identifiers

- ▶ **Identifiers are user defined names**
- ▶ **An identifier consists of letters and digits**



Identifiers

- ▶ Identifiers are user defined names
- ▶ An identifier consists of letters and digits
- ▶ Both uppercase and lowercase letters are permitted



Identifiers

- ▶ Identifiers are user defined names
- ▶ An identifier consists of letters and digits
- ▶ Both uppercase and lowercase letters are permitted
- ▶ First character must be an alphabet or underscore



Summary

- ▶ **Data types**
 - ▶ eg. int, double, float etc.
- ▶ **Variables**
 - ▶ eg. `int a = 2;`
- ▶ **Identifiers**
 - ▶ eg. `printf()`
- ▶ **Constant**
 - ▶ eg. `double const b = 4;`



Assignment

- ▶ Write a C program to calculate the simple interest
- ▶ **Hint:**
 $\text{principal} * \text{rate} * \text{time} / 100$



About the Spoken Tutorial Project

- ▶ Watch the video available at http://spoken-tutorial.org/What_is_a_Spoken_Tutorial
- ▶ It summarises the Spoken Tutorial project
- ▶ If you do not have good bandwidth, you can download and watch it



Spoken Tutorial Workshops

The Spoken Tutorial Project Team

- ▶ Conducts workshops using spoken tutorials
- ▶ Gives certificates to those who pass an online test
- ▶ For more details, please write to contact@spoken-tutorial.org



Acknowledgements

- ▶ **Spoken Tutorial Project is a part of the Talk to a Teacher project**
- ▶ **It is supported by the National Mission on Education through ICT, MHRD, Government of India**
- ▶ **More information on this Mission is available at:**

<http://spoken-tutorial.org/NMEICT-Intro>

