

MIT-IIT Robotics Program

Getting Started, Compiling, Running, I/O, Variables, Expressions

Amartya Shankha Biswas

May 28, 2017

Outline

- 1 Introduction
 - Curriculum
- 2 Environment
 - Terminal and Basic Commands
 - Editor
- 3 Basics
 - Input and Output
 - Variables
 - Expressions
 - Comments
 - Example
- 4 Exercises

- IIT Kharagpur
 - Mayank Bhushan
 - Sudarshan Sharma
 - Manish Agarwal
 - Sayan Sinha
 - Mehul Nirala
 - Rahul Kumar
- MIT
 - Amartya Shankha Biswas
 - Maya Nasr
 - Virup Gubba

Outline

- 1 Introduction
 - Curriculum
- 2 Environment
 - Terminal and Basic Commands
 - Editor
- 3 Basics
 - Input and Output
 - Variables
 - Expressions
 - Comments
 - Example
- 4 Exercises

- Environment
 - Ubuntu Linux
 - Using the Terminal
 - Compiling and Running C++ Code
- Variables and Data Types
- Booleans, Comparison and Logic Flow
- Arrays and Loops
- Functions and Recursion

Advanced C++

Subject to change, based on feedback.

We will decide which of these topics to teach, based on your feedback.

- Graphics and Simulation
 - 2D Graphics
 - Physics Simulaton and Video Games
- Object Oriented Programming
- Algorithms
 - Searching
 - Sorting
- Standard Template Library
 - Set
 - Map

Microcontroller Programming

Working with a tiny computer

- Control Theory (PID controller)
- Programming an Arduino
- Using a Resistive Touchscreen
- Driving Servo Motors

Final Project

Ball and Plate Balancing System

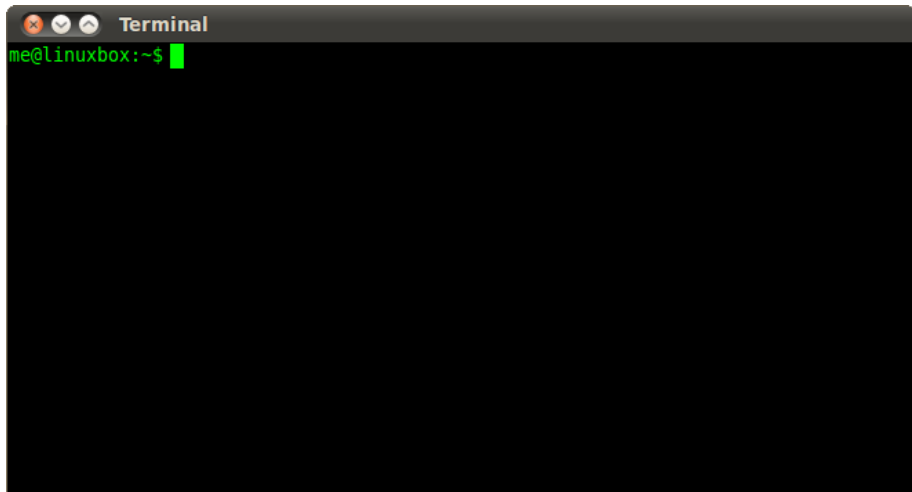
Outline

- 1 Introduction
 - Curriculum
- 2 Environment
 - Terminal and Basic Commands
 - Editor
- 3 Basics
 - Input and Output
 - Variables
 - Expressions
 - Comments
 - Example
- 4 Exercises

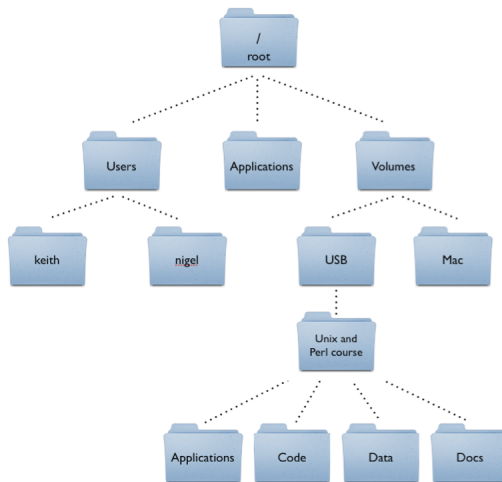
Terminal

Opening Terminal

Open a terminal window by pressing **Ctrl+Alt+T**



Directory Structure



Terminal Navigation

- Type a command in the terminal, and press **Enter**
- Make a new directory (folder) –

```
mkdir <directory_name>
```

- Enter a directory –

```
cd <directory_name>
```

- List contents of directory –

```
ls
```

Outline

- 1 Introduction
 - Curriculum
- 2 Environment
 - Terminal and Basic Commands
 - Editor
- 3 Basics
 - Input and Output
 - Variables
 - Expressions
 - Comments
 - Example
- 4 Exercises

- Write your code in a text editor
 - Notepad, Vim, Emacs, etc.

Text Editor

- Write your code in a text editor
 - Notepad, Vim, Emacs, etc.
- We will use **gedit**

Text Editor

- Write your code in a text editor
 - Notepad, Vim, Emacs, etc.
- We will use **gedit**
- Open *gedit*

```
gedit &
```

- Open a file with *gedit*

```
gedit my_file.cpp &
```


- Write your code in a text editor
 - Notepad, Vim, Emacs, etc.
- We will use **gedit**
- Open *gedit*

```
gedit &
```

- Open a file with *gedit*

```
gedit my_file.cpp &
```

- Note that the file should be present in your current directory
- If file doesn't exist, it will be created

Outline

- 1 Introduction
 - Curriculum
- 2 Environment
 - Terminal and Basic Commands
 - Editor
- 3 Basics
 - **Input and Output**
 - Variables
 - Expressions
 - Comments
 - Example
- 4 Exercises

Hello World !

The First C++ Program

```
#include <iostream>

using namespace std;

int main() {

    cout << "Hello World !" << endl;

    return 0;

}
```

Compiling and Running

- Use the following command to compile –

```
g++ -Wall hello_world.cpp
```

Compiling and Running

- Use the following command to compile –

```
g++ -Wall hello_world.cpp
```

- This should create a file called **a.out** in your current directory.
- Use the following command to run the executable –

```
./a.out
```

Taking Input from User

```
#include <iostream>

using namespace std;

int main() {

    int N;

    cout << "Enter a number: ";
    cin >> N;
    cout << "You entered the number " << N << endl;

    return 0;

}
```

- Statements are terminated by semicolons

```
cout << "Hello World !";
```

- Statements are terminated by semicolons

```
cout << "Hello World !";
```

- A block is several statements inside curly brackets

```
{  
    int N;  
    cin >> N;  
    cout << N << endl;  
    return 0;  
}
```


Try it out

Download Code

Outline

- 1 Introduction
 - Curriculum
- 2 Environment
 - Terminal and Basic Commands
 - Editor
- 3 Basics
 - Input and Output
 - **Variables**
 - Expressions
 - Comments
 - Example
- 4 Exercises

What is a Variable ?

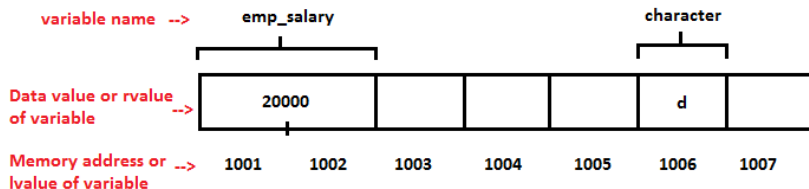
A variable is a **name** that refers to a memory location.

Memory Location

- Name
- Address (L-value)
- Content (R-value)

Memory Location

- Name
- Address (L-value)
- Content (R-value)



`emp_salary` = 2 boxes , as integer takes 2 bytes

`character` = 1 box , as char takes 1 byte only

Data Types

Type	Keyword	Examples
Integer	int	0, -5, 43, 6
Floating point	float	2.5, -0.3, 0.0012, 1.0
Double Floating point	double	0.5, 9.1, -0.7, 7.0

- Variable contains Garbage Value (Un-initialized)

```
int a;
```

Variable Declaration

- Variable contains Garbage Value (Un-initialized)

```
int a;
```

- Initializing with a value.

```
double b = 5.0;
```


Variable Declaration

- Variable contains Garbage Value (Un-initialized)

```
int a;
```

- Initializing with a value.

```
double b = 5.0;
```

- Declare multiple variables

```
float a, b, c, d;
```

Variable Assignment

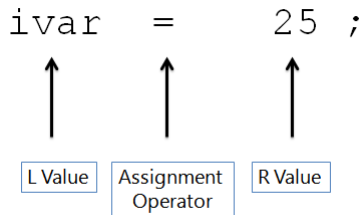
NOT the same as "equals"

Left side is name. Right side is value.

$$a = 5;$$
$$b = a + 5;$$

We "**assign**" a value to a variable.

Variable Assignment



`peri = 2 * (1 + b);`

The diagram illustrates the components of the assignment statement `peri = 2 * (1 + b);`. A dashed arrow points from the label 'l-value' to the variable 'peri'. Another dashed arrow points from the label 'r-value' to the expression '2 * (1 + b)', which is enclosed in a dashed bracket.

Outline

- 1 Introduction
 - Curriculum
- 2 Environment
 - Terminal and Basic Commands
 - Editor
- 3 **Basics**
 - Input and Output
 - Variables
 - **Expressions**
 - Comments
 - Example
- 4 Exercises

Arithmetic in C++

How to make a Calculator

Operation	Symbol	Expression	Equivalent To
Addition	+	$a + b$	—
Subtraction	—	$a - b$	—
Multiplication	*	$a * b$	—
Division	/	a / b	—
Remainder	%	$a \% b$	—

Arithmetic in C++

How to make a Calculator

Operation	Symbol	Expression	Equivalent To
Addition	+	$a + b$	—
Subtraction	—	$a - b$	—
Multiplication	*	$a * b$	—
Division	/	a / b	—
Remainder	%	$a \% b$	—
Increment	++	$++a$	$a = a + 1$
Decrement	--	$--a$	$a = a - 1$

Order of Operations

- Brackets ()
- $*$, $/$
- $+$, $-$

Order of Operations

- Brackets ()
- *, /
- +, -

Using Brackets to Define Order of Operations

$$\begin{aligned} & ((a+b)*(c+d))/5 \\ & (a/b)*(c/d) \end{aligned}$$

Order of Operations

- Brackets ()
- *, /
- +, -

Using Brackets to Define Order of Operations

$((a+b)*(c+d))/5$
 $(a/b)*(c/d)$

Example

```
float pi=3.14159, R=5.0;  
float volume = (4.0/3.0)*pi*R*R*R;
```

- What is the value of the expression $(5/2)$?

Integer Division

- What is the value of the expression $(5/2)$?
 - Answer: 2

Integer Division

- What is the value of the expression $(5/2)$?
 - Answer: 2
- How to do actual division ?
 - $(5.0/2)$
 - $(5/2.0)$

Integer Division

- What is the value of the expression $(5/2)$?
 - Answer: 2
- How to do actual division ?
 - $(5.0/2)$
 - $(5/2.0)$
- If you are dealing with variables – $(x/2)$
 - $(x * 1.0/2)$
 - $(x/2.0)$

Expression	Equivalent To
$a += b$	$a = a + b$
$a -= b$	$a = a - b$
$a *= b$	$a = a * b$
$a /= b$	$a = a / b$
$a \% = b$	$a = a \% b$

Outline

- 1 Introduction
 - Curriculum
- 2 Environment
 - Terminal and Basic Commands
 - Editor
- 3 Basics
 - Input and Output
 - Variables
 - Expressions
 - **Comments**
 - Example
- 4 Exercises

Comments

Explaining what your code does

Anything following `//` will be ignored.

Single Line Comment

```
a += 5; //Adding 5 to the value of a
```

Anything between `/*` and `*/` will be ignored.

Multi Line Comment

```
int N = 0;  
/* This is a comment  
that spans  
multiple lines */  
cin >> N;
```


Outline

- 1 Introduction
 - Curriculum
- 2 Environment
 - Terminal and Basic Commands
 - Editor
- 3 Basics
 - Input and Output
 - Variables
 - Expressions
 - Comments
 - Example
- 4 Exercises

Computing Squares

Putting it All Together

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    double N, square;
```

```
    cout << "Enter a number: ";
```

```
    cin >> N;           //Take input from user
```

```
    square = N*N;       //Compute Square
```

```
    cout << "The square of " << N << " is " << square << endl;
```

```
    return 0;
```

```
}
```

Computing Cubes

```
#include <iostream>

using namespace std;

int main() {

    double N, cube;

    cout << "Enter a number: ";
    cin >> N;           //Take input from user
    cube = N*N*N;       //Compute cube
    cout << "The cube of " << N << " is " << cube << endl;

    return 0;

}
```

Coding Practice

Area of Circle

Write a program that takes as input the length of the radius of a circle, and outputs its area.

Sum of First N Natural Numbers

Write a program that takes as input an integer N , and computes the sum $(1 + 2 + 3 + \dots + N)$. You may use the fact that $1 + 2 + \dots + N = \frac{N \cdot (N+1)}{2}$.

Write a program that takes as input an integer N , and computes the sum of the last three digits of N . If there are less than three digits, just sum all of them.

If you are familiar with C/C++

Write a program that takes as input an integer N , and computes the following sum.

$$\frac{6}{1^2} + \frac{6}{2^2} + \frac{6}{3^2} + \frac{6}{4^2} + \dots + \frac{6}{N^2}$$