C++

PROGRAMMING LANGUAGE

L01 – VARIABLES

Mohammad Shaker
mohammadshaker.com
@ZGTRShaker
2010, 11, 12, 13, 14

C/ C++/ C++.NET/ C#

C/ C++/ C++.NET/ C#

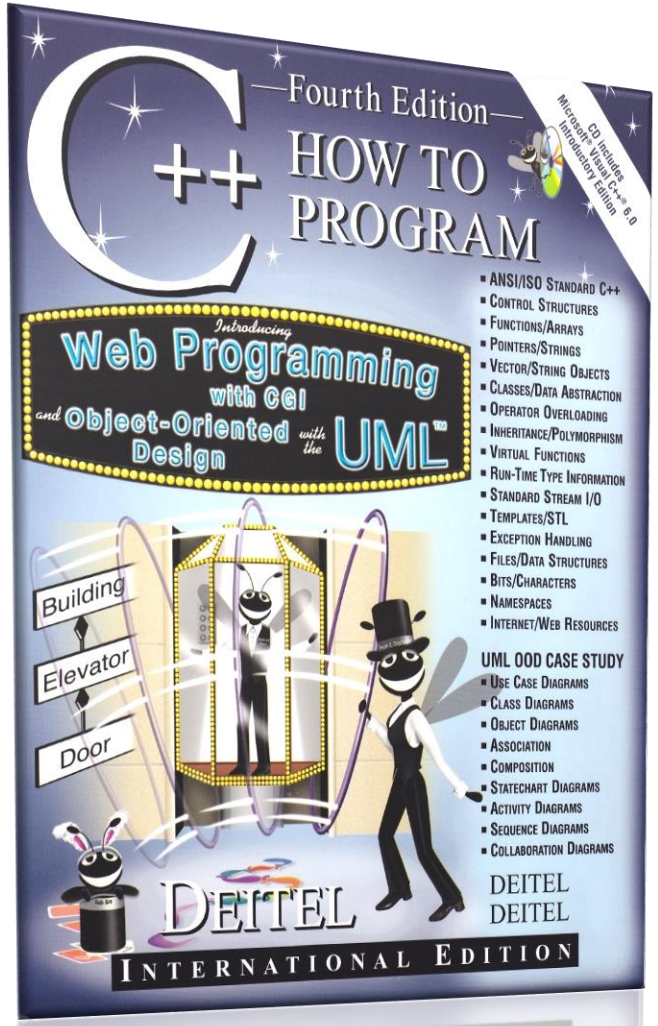
Who Are the Guys?

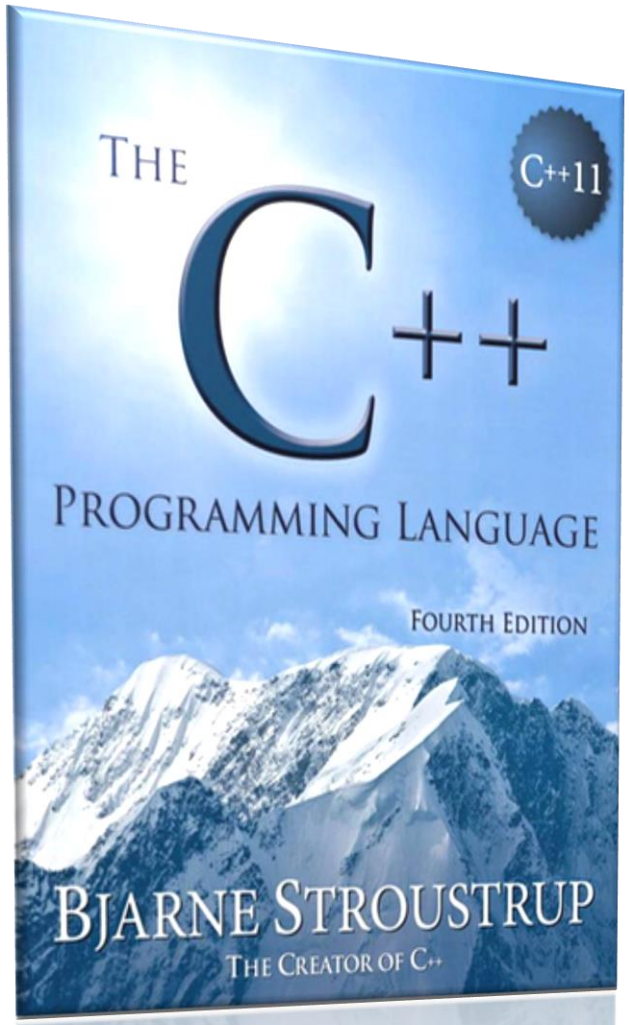
DENNIS RITCHIE


BJARNE
STROUSTRUP

Resources

- Books
  - Deitel – C++ How to program
  - The C++ Programming Language (Not for complete starters)
- Bunsh of good websites
  - www.cplusplus.com
  - www.msdn.com
- msdn awesome library
- `stackoverflow.com` and `google` are always your best programming buddies


C++
Fourth Edition
HOW TO PROGRAM
CD Includes Microsoft Visual C++ 6.0 Introductory Edition
Introducing Web Programming with CGI and Object-Oriented Design with the UML™
ANSI/ISO Standard C++
Control Structures
Functions/Arrays
Pointers/Strings
Vector/String Objects
Classes/Data Abstraction
Operator Overloading
Inheritance/Polymorphism
Virtual Functions
Run-Time Type Information
Standard Stream I/O
Templates/STL
Exception Handling
Files/Data Structures
Bits/Characters
Namespaces
Internet/Web Resources
UML OOD CASE STUDY
Use Case Diagrams
Class Diagrams
Object Diagrams
Association
Composition
Statechart Diagrams
Activity Diagrams
Sequence Diagrams
Collaboration Diagrams
Building
Elevator
Door
DEITEL
DEITEL
DEITEL
INTERNATIONAL EDITION


THE
C++11
C++
PROGRAMMING LANGUAGE
FOURTH EDITION
BJARNE STROUSTRUP
THE CREATOR OF C++

What You Will Learn

- Concepts you already know
  - Variables
  - Control Structure
  - Functions
  - Arrays/ Pointers/ Strings
  - Structs

- New concepts
  - Classes (OOP)
  - Inheritance (OOP)
  - Polymorphism (OOP)
  - Template
  - STL
  - Exception Handling
  - File Processing

Programming Languages War

Aug 2014	Aug 2013	Change	Programming Language	Ratings	Change
1	2	▲	C	16.401%	+0.43%
2	1	▼	Java	14.984%	-0.99%
3	4	▲	Objective-C	9.552%	+1.47%
4	3	▼	C++	4.695%	-4.68%
5	7	▲	Basic	3.635%	-0.24%
6	6		C#	3.409%	-2.71%
7	8	▲	Python	3.121%	-0.48%
8	5	▼	PHP	2.864%	-3.83%
9	11	▲	Perl	2.218%	+0.18%
10	9	▼	JavaScript	2.172%	+0.08%
11	-	⇈	Visual Basic	2.014%	+2.01%
12	13	▲	Visual Basic .NET	1.310%	-0.01%
13	10	▼	Ruby	1.242%	-0.83%

August 2014, Source: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Moore’s Law and Other Stuff

C++ Preferences & features

- Moore's Law
  - Computer processing power can be doubled every 18-24 months
  - Software with Hardware improves together
    - More complicated Hardware
    - More advanced Software
  - Now, we are in generation that we put the "Moore's Law" behind us!

C++ Preferences & features

- Two kind of Computer programming paradigms:
  - Imperative
    - Procedural (Pascal, C, php, etc)
    - Object-Oriented (C++, C#, Java, ASP.NET, etc)
  - Declarative
    - Functional (Pascal, C, php, etc)
    - Logic (Prolog, etc)

---

* Note the need to C++ after inventing C

** Note that the Procedural \ Functional programming paradigms are still used till our present time and has its unique features

C \ C++

- The "C" Programming language is a modular one
- Why C++ then?
  - Objects, OOP
  - C \ C++ are portables ones

- Comparison:
  - C: is action Oriented
    - Procedural
  - C++: is object oriented
    - Compiler checking
    - Extensible language
      - Class
    - Reusable

Where you can find C++?

WHERE YOU CAN FIND C++?

PRETTY EVERYWHERE!

(TAKE A LOOK HERE: HTTP://WWW.STROUSTRUP.COM/APPLICATIONS.HTML)

MICROSOFT, OFFICE
GOOGLE
HP
AMAZON
ADOBE
MOZILLA
MYSQL
INTEL
NOKIA
SUN
BLOOMBERG
GAME ENGINES

POWERED BY

UNREAL
ENGINE


Infiltrator
File Edit Window Help
Tools
L Essentials
Save
Content
View
World Settings
Level Blueprint
Matinee
Build
Simulate
Play In
Details
Levels
Select an object to view det
Scene Outliner
show
Actor
Info
P_Show_Impact_Metal_I (Emitter)
P_Show_Impact_Metal_I (Emitter)
P_Show_Impact_Metal_I (Emitter)
P_Show_Impact_Metal_I (Emitter)
P_Show_Impact_Purple (Emitter)
P_Show_Impact_WhiteLi (Emitter)
P_Show_Impact_WhiteLi (Emitter)
P_Show_Impacts_Water (Emitter)
P_Show_Impacts_Water (Emitter)
P_Show_ImpactWater_L (Emitter)
P_Show_Robot_Lights (Emitter)
Show_Char_ (SkeletalMeshActor)
P_Show_Weap_Muz (Emitter)
Show_Char_ (SkeletalMeshActor)
P_Show_Muz_2 (Emitter)
Show_Char_ (SkeletalMeshActor)
Show_Water_Cam (CameraActor)
Show_Water_Cam (CameraActor)
SK_Show_Bl (SkeletalMeshActor)
P_Show_Muzzle_Bot (Emitter)
Showing 21 of 3719 actors
View Options
Emitters
Particle Emitter
Particle Emitter
BurstLight
LoopLight
Particle Emitter
Heat
Mesh Data
Required
Spawn
Lifetime
Initial Size
Initial Color
Initial Velocity
Size By Life
Content Browser
Game
New Asset
Filters
P_MeshSplash_Demo_02
P_Muzzle_Flash_1
10 items
View Options

Enough talk!
Let's get into the

Action!

The Structure of a C++ Program

Structure of C++ program

```
#include <iostream>

void main ()
{

}
```

```
#include <iostream>
using namespace::std; // Note the new namespace

void main()
{
          cout << "we're having fun!";
}
```

```
#include <iostream>

int main ()
{
          // indicate successful termination
          return 0;
}
```

```
#include <iostream>
using namespace::std; // Note the new namespace

int main ()
{
          cout << "we're having fun!";
          return 0; // indicate successful termination
}
```

Structure of C++ program

```
#include <iostream>
using namespace::std; // Note the namespace
void main()
{cout << "we're having fun!";}
```

```
#include <iostream>
using namespace::std;
void main()
{
        cout << "We're having fun!";
        cout << "I need to eat!:D ";
}
```

```
We're having fun!I need to eat!:D
```

```
#include <iostream>
using namespace::std; // Note the namespace
int main ()
{
        cout << "we're having fun!";  return 0;
}
```

```
#include <iostream>
using namespace::std;

void main()
{
        cout << "we're having fun!" << endl;
        cout << "I need to eat!:D " << endl;
}
```

```
we're having fun!
I need to eat!:D
```

Structure of C++ program

```
#include <iostream>
using namespace::std;

void main()
{
Cout<<"we're having fun!"
<< endl<<"I need to eat!:D " << endl;
}
```

```
we're having fun!
I need to eat!:D
```

```
#include <iostream>
using namespace::std;

void main()
{
Cout<<"we're having fun!"<<endl<<"I need to eat!:D "
<< endl;
}
```

```
we're having fun!
I need to eat!:D
```

Structure of C++ program

```
#include <iostream>
using namespace::std;

void main()
{
          cout << "we're having fun!" << "\n" <<  "I
need to eat!:D "
          << "\n";
}
```

```
we're having fun!
I need to eat!:D
```

```
#include <iostream>
using namespace::std;

void main()
{
  cout << "we're having fun! \n I need to eat!:D \n";
}
```

```
we're having fun!
I need to eat!:D
```

`\n` is the same as **`endl`**

Structure of C++ program

```
#include <iostream>
using namespace::std;

void main()
{
          cout << "foo"; // this is a line!
}
```

Compile and run

```
#include <iostream>
using namespace::std;

void main()
{
          cout << "foo"; /* this is a line! */
}
```

Compile and run

Structure of C++ program

```
#include <iostream>
using namespace::std;

void main()
{
          cout << "foo"; // this is
          a line!
}
```

Compiler error

```
#include <iostream>
using namespace::std;

void main()
{
          cout << "foo"; /* this is not a single
          line! */
}
```

Compile and run

Comments

```
// comment
```

For a one line

```
/*
        comments
                        */
```

For one line or more (multi lines)

Escape code

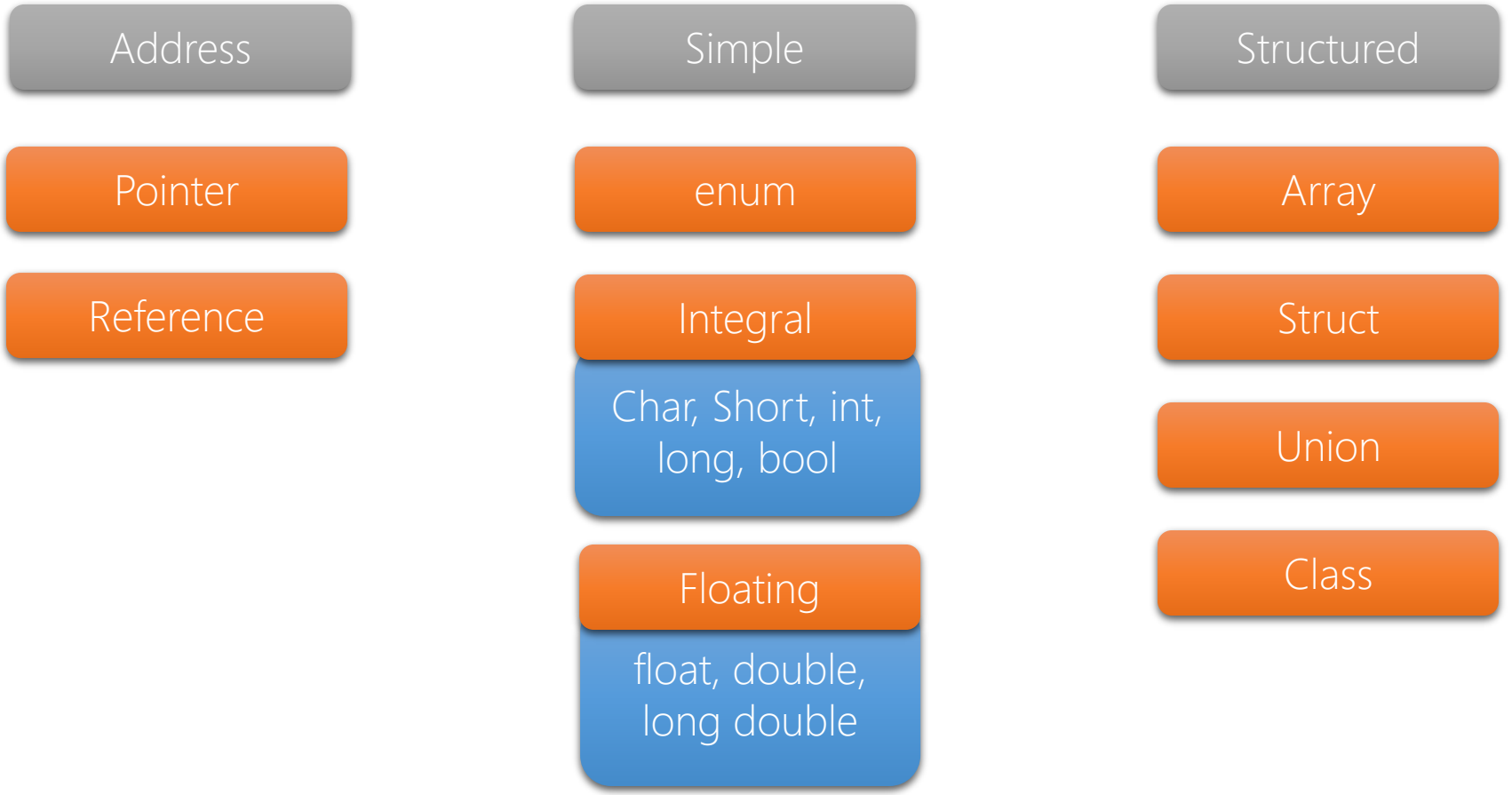
	Escape Sequence	Description
\n	Newline	Cursor moves to the beginning of the next line
\t	Tab	Cursor moves to the next tab stop
\ b	Backspace	Cursor moves one space to the left
\r	Return	Cursor moves to the beginning of the current line (not the next line)
\\	Backslash	Backslash is printed
\'	Single quotation	Single quotation mark is printed
\"	Double quotation	Double quotation mark is printed

I/O Stream

- Keyboard / Screen

Variables

C++ data types


Address
Pointer
Reference
Simple
enum
Integral
Char, Short, int, long, bool
Floating
float, double, long double
Structured
Array
Struct
Union
Class

Variables

- Every variable has:
  - Name
  - Type
  - Size
  - Value

- Data Types:
  - Integer, Double, float, char

Variables

```
#include <iostream>
using namespace::std;

void main()
{
          int i;
          float j;
}
```

```
Compile and Run
```

```
#include <iostream>
using namespace::std;

void main()
{
          int i;
          int j;
}
```

```
Compile and Run
```

```
#include <iostream>
using namespace::std;

void main()
{
          int i, j;
}
```

```
Compile and Run
```

Variables

```
#include <iostream>
using namespace::std;
 void main()
{
          int i, j;
          i = 0;
          j = 4;
}
```

Compile and Run

```
#include <iostream>
using namespace::std;

void main()
{
          int i = 0, j = 4;
}
```

Compile and Run

Variables

```
#include <iostream>
using namespace::std;

void main()
{
            int i;
            cin >> i;
            cout << " i = " << I;
}
```

```
Compiler error, undeclared "I" identifier
```

C++ is case sensitive

```
#include <iostream>
using namespace::std;

void main()
{
            int I;
            cout << I;
}
```

Runtime Error – Visual 2010

Variable used With out initiating first

Variables

```
#include <iostream>
using namespace::std;

void main()
{
          int i, j;
          cin >> i >> j;
          int sum = i + j;
          cout << sum;
}
```

```
#include <iostream>
using namespace::std;

void main()
{
          int i, j;
          cin >> i >> j;
          int sum = i + j;
          cout << " sum = " << sum;
}
```

Variables

```
// Operating with variables
#include <iostream>
using namespace std;
int main ()
{
  // declaring variables:
  int a, b;
  int result;

  // process:
  a = 5;  b = 2;  a = a + 1;
  result = a - b;

  // print out the result:
  cout << result;

  // terminate the program:
  return 0;
}
```

Variables

```
#include <iostream>
using namespace std;

void main()
{
            int i = 0;
            i = i + 1;
            cout << i;
}
```

```
1
```

```
#include <iostream>
using namespace std;

void main()
{
            int i = 0;
            i += 1;
            cout << i;
}
```

```
1
```

```
#include <iostream>
using namespace std;

void main()
{
            int i = 0;
            i++;
            cout << i;
}
```

```
1
```

Variables

```
#include <iostream>
using namespace std;

void main()
{
            int i = 0;
            i--;
            cout << i;
}
```

```
-1
```

```
#include <iostream>
using namespace std;

void main()
{
            int i, j;
            i++;
            j--;
            cout << i << j;
```

```
Runtime Error - Visual Studio 2010
```

Variables

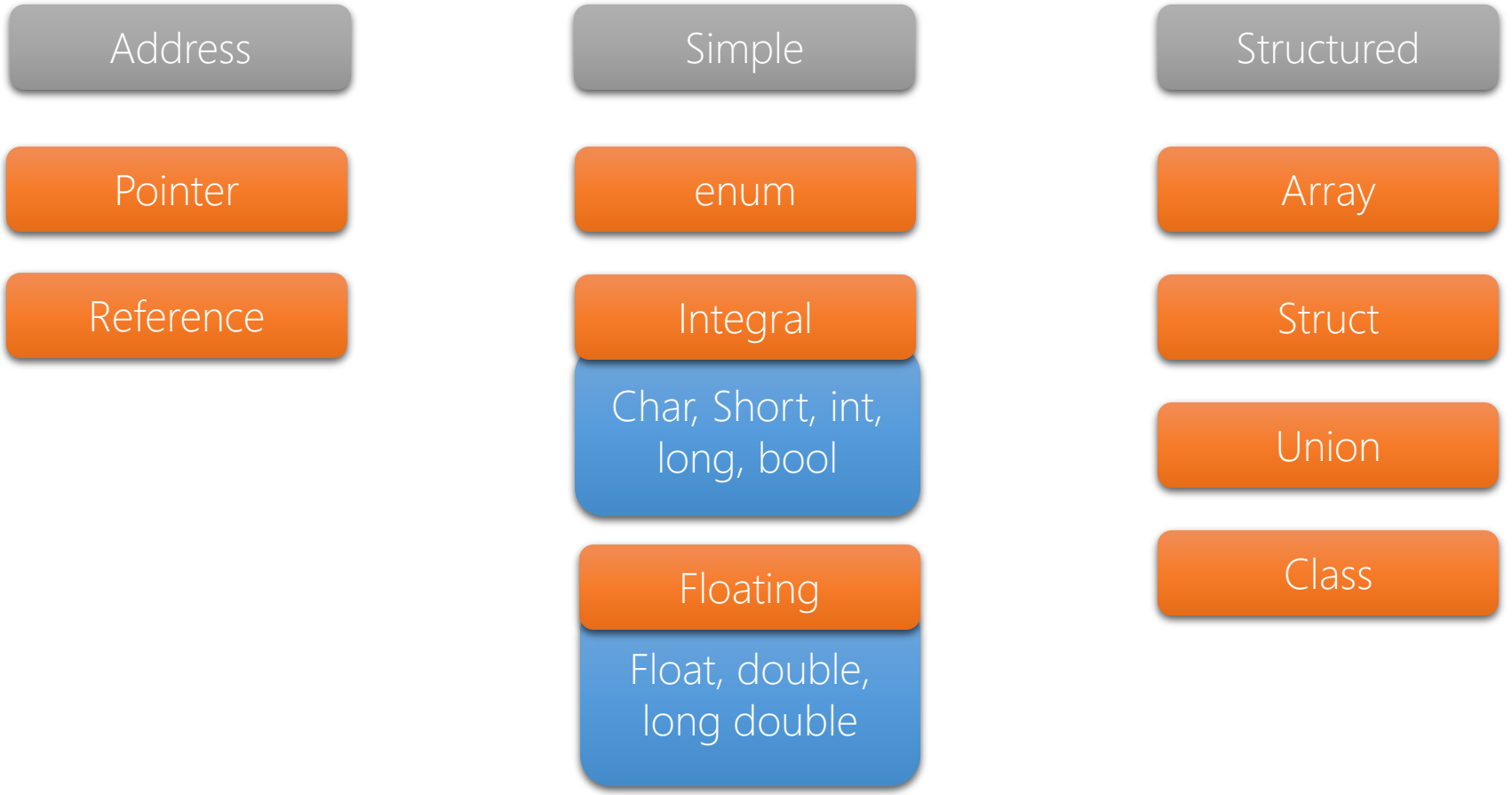
```
#include <iostream>
using namespace::std;

void main()
{
            int i, j;
            cin >> i;
            cout << endl;
            cout << "i = " << i << endl << "\t";
            cin >> j;
            cout << "j = \n" << j;
            cout << "_____" << endl;
            cout << "j+i = " << j+i  << "\n";
            cout << "2*i = " << 2*i  << "\n";
}
```

```
2

i = 2
3
j =
3_____
j+i = 5
2*i = 4
Press any key to continue
```

C++ data types


Address
Pointer
Reference
Simple
enum
Integral
Char, Short, int, long, bool
Floating
Float, double, long double
Structured
Array
Struct
Union
Class

integral

Integral

- `char, short, int, long, bool`
- `char`
  - Used to represent character such as:
    - Letters
    - Digits
    - Special symbols
      - ' + ', ' & ', ' $ ', ' * '
  - Each character is enclosed with single quote mark ' ' and not double ones " "
  - Space is represented by ' ' with space between them.
  - ASCII & EBCDIC

- `bool`
  - Watch out that the "Boolean" type is an integral one!
  - bool
    - false = 0, true = any other number

integral

```
#include <iostream>
using namespace::std;

void main()
{
            char c1 = 'd', c2;
            cout << c2 << c1 << endl;

}
```

```
 d
```

```
void main()
{
            bool b1, b2;
            if (3 <= 2)
            {
                        b1 = false;
                        b2 = true;
            }
            else
            {
                        b1 = 53;
            }
            cout << b1 << " - " << b2 << endl;
}
```

```
1 - 0
```

Note that:

- The default value for a char is **NULL** represented as **SPACE**' ' but it's not a space, it's a **NULL**!
- The char has ' ' and not " "

Note that:

- A numeric value is printed when printing a "boolean"
- The default value for bool type is false (0)

Floating point data types

- float, double:
  - float 4 bytes / double 8 bytes
    - float has a single precision
    - double has a double precision
  - double = long double (in new compilers)
  - The size of float, double, long double are machine dependent.

integral

```
#include<iostream>
using namespace::std;

void main()
{
            double d = 0.4;
            cout << d << endl;
            system("pause");
}
```

```
0.4
```

```
#include<iostream>
using namespace::std;

void main()
{
            double d =.0;
            cout << d << endl;
            system("pause");
}
```

```
0
```

```
#include<iostream>
using namespace::std;

void main()
{
            double d = 0.0;
            cout << d << endl;
            system("pause");
}
```

```
0
```

```
#include<iostream>
using namespace::std;

void main()
{
            double d = 0.0;
            cout << d << endl;
            system("pause");
}
```

```
0
```

integral

```
#include<iostream>
using namespace::std;

void main()
{
            float f = 1.2;
            cout << f << endl;
            system("pause");
}
```

```
1.2
```

```
#include<iostream>
using namespace::std;

void main()
{
            float f = 1.2f;
            cout << f << endl;
            system("pause");
}
```

```
1.2
```

Constants, `const`

Constants

- A Constant:
  - Any expression that has a “fixed” value

- 3 kind of constants:
    - Integer Numbers
    - Floating-Point Numbers
    - Characters & Strings

Constants

- Integer Numbers
  - `1225 // Decimal`
  - `-982 // Decimal`
  - `05356 // Octal!`
    - Octal numbers are preceded by 0
  - `0x3c // Hexadecimal`
    - Hexadecimal numbers are preceded by 0x

- Floating Numbers
  - Decimal
  - Exponent

  Examples:
  - `5.0 // 5.0 (double)`
  - `5.0f // 5.0 (float)`
  - `45.556779 // 45.556779`
  - `8.36e18 // 8.36 x 10^18`
  - `8.36e-18 // 8.36 x 10^-18`

- Characters and Strings
  - `'Z' //Char – Single Character`
  - `'M' //Char – Single Character`
  - `"Where's the cat?" //String – Several Character`
  - `"I just don't know!" //String – Several Character`
  - `"c" //String – One Character`

integral

```
#include <iostream>
using namespace::std;
#define PI 3.14;                          // No “=” Sign
#define MyTab '\t'                        // No “=” Sign
#define PonPon ":D"                       // No “=” Sign


void main()
{
}
```

```
#include <iostream>
using namespace::std;

void main()
{
      #define PI 3.14;
      #define MyTab '\t'
      #define PonPon ":D"
}
```

integral

```
#include <iostream>
using namespace::std;
#define MyTab '\t'

void main()
{
          #define PI 3.14;
          float Radius = 0;
          cout << "Enter the Radius" << endl;
          cin >> Radius;
          float Circle = PI;
          Circle = Circle * 2 * Radius;
          cout << "The perimeter of the Circle = "<< Circle
          << MyTab;
}
```

```
Enter the Radius

3.2

The perimeter of the Circle = 20.096    Press any key to continue
```

integral

```
#include <iostream>
using namespace::std;
#define MyTab '\t'
#define PI 5;
void main()
{
          #define PI 3.14;
          float Radius = 0;
          cout << "Enter the Radius" << endl;
          cin >> Radius;
          float Circle = PI;
          Circle = Circle * 2 * Radius;
          cout << "The perimeter of the Circle = "<< Circle
          << MyTab;
}
```

```
Enter the Radius

3.2

The perimeter of the Circle = 20.096    Press any key to continue
```

integral

```
#include <iostream>
using namespace::std;


void main()
{
          #define PI 3.14;
          float Radius = 0;
          cout << "Enter the Radius" << endl;
          cin >> Radius;
          float Circle = 2 * PI * Radius;
          cout << "The perimeter of the Circle = " << Circle;
}
```

```
Illegal Indirection 2 * PI * Radius
```

integral

```
#include <iostream>
using namespace::std;
const int x = 20;

void main()
{
          const int y = 90;
}
```

```
Compile and run
```

```
#include <iostream>
using namespace::std;
const int x = 20;

void main()
{
          const y = 90;
}
```

```
2005 Compiler: Compile & Run
int assumed for const type when neglecting the
type
```

```
const char Me = 'M';
const int Height = 5;
const char MyCharTab = '\t';              // Char
tab
const char *MyStringTab = "\t";
          // String tab

void main()
{
          cout << MyStringTab;
}
```

```
          Press any key to continue
```

```
const char MyCharTab = '\t';                    // Char tab
const char *MyStringTab = "\t";                 // String
tab
#define MyStringTab "\t";                       // String
tab

void main()
{
          cout << MyStringTab;
}
```

```
          Press any key to continue
```


Let’s Crack Some Code

Code Cracking

```
#include <iostream>

int main ()
{
        return 2;
}
```

```
// Will compile & Run!
// Any number will do the job
```

```
#include <iostream>

int main ()
{

}
```

```
compile & Run
```

```
#include <iostream>

int main ()
{
        return;
}
```

```
// compiler error
// must have a return value
```

```
#include <iostream>
using namespace::std;

void main()
{
        cout << "foo"; /* this is a line!
}
```

```
Compiler error, missing */
```

Code Cracking

```
#include <iostream>
using namespace::std;

void main()
{
        cout >> "foo"; /* this is a line! */
}
```

Compiler error, <<

```
#include <iostream>

int main()
{
        int i_j;
        return -0001223333;
}
```

compile & Run

```
#include <iostream>
using namespace::std;

void main()
{
        int i, int j;
}
```

Compiler error

```
#include <iostream>

Int main()
{
        Int i_j;
        return -12323333;
}
```

Compiler error "Int" is capital letter

Code Cracking

```
#include <iostream>
using namespace::std;

void main()
{
          bool b1, b2;
          if (b1 >= b2)
          {
                    b1 = 0;
                    b2 = 1.3;
          }
          cout << b1 << b2 << endl;
}
```

```
01
```

```
#include <iostream>
using namespace::std;

void main()
{
          bool b1, b2;
          if (b1 >= b2)
          {
                    b1 = not(0);
                    b2 = 1.3;
          }
          cout << b1 << b2 << endl;
}
```

```
Compiler error,
no such keyword as "not"
it is "!".
```

Code Cracking

```
#include <iostream>
using namespace::std;

void main()
{
          bool b1, b2;
          if (b1 >= b2)
          {
                    b1 =!(0);
                    b2 = 0;
          }
          cout << b1 << b2 << endl;
}
```

```
10
```

```
#include <iostream>
using namespace::std;

void main()
{
          bool b1=1, b2;
          if (b1!= b2)
          {
                    b1 =!(1-2312);
                    b2 =!!b1;
          }
          cout << b1 << b2 << endl;
}
```

```
00
```

Code Cracking

```
#include <iostream>
using namespace::std;

void main()
{
          bool b1, b2;
          if (b1 >=!(23423))
          {
                    b1 =!(1-1231233);
                    b2 =!!b1;
          }
          cout << b1 << b2 << endl;
}
```

```
00
```

```
#include <iostream>
using namespace::std;

void main()
{
          double b1, b2;
          if (b1!=!(34))
          {
                    b1 =!(1-1231233);
                    b2 =!b2 + b1;
          }
          cout << b1 << b2 << endl;
}
```

```
00
```

Code Cracking

```
#include <iostream>
using namespace::std;
 void main()
{
          double b1, b2;
          bool f;
          if (b1!=!(f))
          {
                    b1 =!(1-1231233);
                    b2 =!b2 + b1;
          }
          cout << b1 << b2 << endl;
}
```

```
01
```

```
#include <iostream>
using namespace::std;
 void main()
{
          double i1 = 9.3, i2 = 3.6;
          bool b1, b2;
          if (i1 > i2)
          {          b1 = -1;  b2 = true;          }
          else
          {          b1 = 53;  }
          cout << b1 << "\t -\t" << b2
          <<"\n\n\n\t\t - "<<b2<< endl;;
}
```

```
1         -       1


                  - 1
Press any key to continue
```

Code Cracking

```
#include <iostream>
using namespace::std;

void main()
{
          char c1 = 'd', c2;
          bool b1;
          cout << int(c2) << " - c1\t - " << int(b1) <<
endl;

}
```

```
0 - c1      - 0
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main()
{
          char c1 = "d", c2;
          bool b1;
          cout << int(b1) << endl;

}
```

```
Compiler error " "
```

Code Cracking

```
#include<iostream>
using namespace::std;

void main()
{
	cout << "I'm number 4 or 77, I don't know :D" << endl;
	;
	;
	;
	;
	system("pause");
}
```

```
I'm number 4 or 77, I don't know :D
```

Related Online Courses

Programming Paradigms

C++ Memory Management. LISP and Python

http://see.stanford.edu/see/courseinfo.aspx?coll=2d712634-2bf1-4b55-9a3a-ca9d470755ee

Programming methodology - Java

http://see.stanford.edu/see/courseinfo.aspx?coll=824a47e1-135f-4508-a5aa-866adcae1111

Take a look at my other courses,
Especially the C++.NET
http://www.slideshare.net/ZGTRZGTR/

Keep in touch and let's connect


http://www.mohammadshaker.com


mohammadshakergtr@gmail.com


http://mohammadshakergtr.wordpress.com/


https://de.linkedin.com/pub/mohammad-shaker/30/122/128/


https://twitter.com/ZGTRShaker @ZGTRShaker


http://www.slideshare.net/ZGTRZGTR


https://www.goodreads.com/user/show/11193121-mohammad-shaker


https://plus.google.com/u/0/+MohammadShaker/


https://www.youtube.com/channel/UCvJUfadMoEaZNWdagdMyCRA

HOPE YOU HAVE ENJOYED YOUR FIRST CLASS

SEE YOU
TOMORROW!