SE116
2014-2015 SPRING

LECTURE 1

by İlker Korkmaz

Before starting:
Exams and Assessment

There are 6 different sections instructed by 3 different lecturers and tutored with the same materials. All registered students will be evaluated via the same exams to be held at the same times.

Grading is to be fair based on the following procedure: Every specific question of the exams will be graded by the same corresponding instructor.

Syllabus:
http://ects.ieu.edu.tr/new/syllabus.php?section=ce.cs.ieu.edu.tr&course_code=SE%20116&lang=en

Review on C

- Remainder notes on some concepts of C
  - algorithm, pseudocode, implementation
  - source code, object code, executable code
  - preprocessor, compiler, linker
  - declaration, definition, initialization
  - keywords of C, reserved words in C
  - function
  - prototype, signature
  - storage classes
  - static versus dynamic memory allocation
  - struct / data structure
  - all above are well-defined in “wikipedia”

How to shift from C to C++

- 1-> You might remember how to implement a code using the concepts of programming with C, which is an imperative language together with the methods of a procedural programming paradigm.

- 2-> Try to understand the concepts of object oriented programming paradigm.

Now, let’s begin with Lecture 1: from C to C++

- The main textbook is “C How to Program, 7th Edition” by Deitel and Deitel.
- This presentation covers the subjects given in Chapter 15 of the textbook.

from C to C++

Namespaces

- C++ provides ***namespaces*** to prevent name conflicts
- Example:

```
namespace X { //X refers to Company X
     int ID;
     //...
}

namespace Y { //Y refers to Company Y
     int ID;
     //...
}
```

using namespaces

- explicitly usage of the variables:

```
#include <iostream>

using X::ID;

int main() {
        ID =1313;
        Y::ID =2626;
        return 0;
}
```

cin / cout

- scanf --> cin
- printf --> cout

```
#include <iostream>

using namespace std;

int main() {

    int val, sum =0;
    cout<< "Program calculates the sum:" <<endl;
    cout<< "Enter a number: ";
    while (cin >> val) {
        sum+=val;
        cout <<"Enter next number:";
    }
    cout<<"Sum of all values: " <<sum << '\n';
    return 0;
}
```

A New Type: string

- ***string*** is an alternative type to C's null terminated arrays of ***char***

```
#include <iostream>

using namespace std;

int main() {
    string s1;
    string s2="Hello";
    string s3 = s2;
    string myString(5,'x');
    s1="This is a string";
    return 0;
}
```

string length

- string st= “Hello”;
- cout << “Length of st is ” <<st.length();
- Length of st is 5

```
#include <iostream>
#include <string>

using namespace std;

int main() {
      string s1;
      string s2="Hello";
      string s3 = s2;
      string myString(5,'x');
      s1="This is a string";
      cout << s2.length();
      cout << myString;
      return 0;
}
```

string operations

- concatenation, modifying, extracting a substring, searching, comparing...

function concepts

- FUNCTIONS
  - prototype ...
  - call by reference (&)...
  - inline functions...
  - default arguments...
  - overloading functions...
  - function signatures...

dynamic storage allocation in C++

- new, new[ ]
  - The ***new*** operator is used to allocate storage dynamically.
  - The ***new[ ]*** operator is used to allocate an array dynamically.
- delete, delete[ ]
  - The ***delete*** operator is used to free storage allocated by ***new.***
  - The ***delete [ ]*** operator is used to free storage allocated by ***new [ ].***
- "**new**" and "**delete**" keywords will also be used with constructors and destructors.

examples to allocate storage

- int * number_ptr;
- .....
- number_ptr= new int ;
- delete number_ptr;

- // alternatively, array allocation approach is as follows:

- int * number_ptr;
- .....
- number_ptr=new int[100] ;
- delete [ ] number_ptr;

struct/class relation

- in C : “***struct***”s are the basic structures..
- in C++ : “***class***”es are the basic structures..

More slides ??

- You may also dissect the slides offered by Dr. Ufuk Çelikkan and Dr. Senem Kumova Metin. All those slides are conceptually similar.