C++ Programming Language

Lecturer: Yuri Nefedov 7th and 8th semesters Lectures: 34 hours (7th semester); 32 hours (8th semester). Seminars: 34 hours (7th semester); 32 hours (8th semester).

Course abstract

Programming is now a routine activity of scientists, be it experimental data processing or theoretical calculations. The increased complexity of experiments in physics – especially in high energy physics – has led to the domination of software written in C++. The concepts which underlie C++ are versatile and very powerful; the language is difficult to learn – especially for beginners. Therefore, the course consists of two parts: the first one presents the C subset of C++ and programming basics. The second part presents the C++ language quite fully; students are acquainted with object-oriented programming.

The seminars are held in parallel with lectures. At the seminars, problems are solved which illustrate the subjects presented at the lectures.

Part 1: C - subset

Lecture 1.

Introduction. History of creation of C and C++. Acquaintance with the language C.

Seminar.

The program "HELLO, WORLD". What is the translator and how to use it.

Lecture 2.

Types of the data. Identificators. Region of visibility. Acquaintance with the operators. arithmetical operators. Hybrids with the assignment. The operators ++ and --.

Seminar.

The type int. A program for acquaintance with the operators.

Lecture 3.

Representation of numbers in the computer. Bit operations. The operator sizeof.

Seminar.

The types char and unsigned int. A program of printing an entire number in the binary representation.

Lecture 4.

Input-output, C-style. Functions of the families printf and scanf.

Seminar,

The type double. A program of printing tables of sines and cosines.

Lecture 5.

Managing of a program. The operators if and goto. Logical operators. Operators of the relation. Ternary operator. The choice operator switch.

Seminar.

A program of counting letters and numbers in an input file.

Lecture 6.

Cycles while, do-while, for. The operators break and continue.

Seminar.

The program of computing a mean value and the dispersion for a distribution.

Lecture 7.

Functions. Parameters and variables. Transmission by a value. Declaration and definition of a function.

Seminar.

Formatting the previous programs as functions.

Lecture 8.

Addresses and indexes. How to avoid troubles with the indexes? Arithmetic of the indexes.

Seminar.

The program of swapping two numbers (swap). The bubble sorting.

Lecture 9.

Arrays. Access to the elements. Transfer of the arrays in parameters. Strings, C-style.

Seminar.

A program for "histogramming" of the length of strings in a file.

Lecture 10.

Arrays and indexes. Multi-dimensional arrays, arrays of indexes. The "real" function main().

Seminar.

The program analyzing its inner parameters. Printing of the environment variables.

Lecture 11.

User types of data in C. Structures. Byte fields. Unifications. Enumeration. The operator typedef.

Seminar.

An improved version of the program for "histogramming" the length of strings in a file using the structures.

Lecture 12.

Dynamical memory, C-style. The universal index void*.

Seminar.

A program using the functions malloc(), calloc(), realloc() and free().

Lecture 13.

The front-end processor. Phases of the translation[s]. Directives of the front-end processor. Macrodefinitions.

Seminar.

Application of the macrodefinitions. Printing of byte size of all built-in types.

Lecture 14.

The front-end processor and a separate translation. The specificators extern, static. Specificators of the type const, volatile.

Seminar.

Compilation of the programs stored in different files. The notion on the "make" file.

Saving programs in a library file. The header file.

Lecture 15.

Survey of the standard library C: Functions of working with the errors, errno. Streams: buffered/non-buffered input-output<stdio.h>.

Seminar.

The program of reading from a file with the verification of correctness of opening and reading.

Lecture 16.

Survey of the standard library C (continuation): Functions of the common use <stdlib.h>. THE mathemarical functions <math.h>. Analysis of the symbols <ctype.h>. Working with the strings <string.h>.

Seminar.

Using the program sort() from the standard library C.

Lecture 17.

Survey of the standard library C (continuation): Lists of arguments of the variable length <stdarg.h>. Working with the date and time <time.h>. The limits <limits.h>, <float.h>. Working with the signals <signal.h>.

Seminar.

The program of printing the current date and time. Computing the time spent by the program.

Part 2: C++

Lecture 1.

Introduction. Recommended literature. Evolution of the programming languages.

Seminar.

The program "HELLO, WORLD" in C++. Translator of the language C++.

Lecture 2.

C++ as an "improved" C. New version of the comments. New syntaxes of reduction of types and initialization. Creation of the variables. Built-in (inline) functions. Unnamed parameters.

Seminar.

Using inline-functions instead of macrodefinitions. The program of printing first 100 simple numbers.

Lecture 3.

C++ as an "expanded" C. The new style of input-output. Name space. Reloading of the functions. Prototypes and signatures.

Seminar.

The program using mixed input-output in C and C++ styles.

Lecture 4.

Links (references). Dynamical memory, the C++-style.

Seminar.

The program of swapping (swap) of two numbers using links. Application of "new" and "delete" for arrays.

Lecture 5.

User types of data: classes. What for one needs classes if there are already structures? Object style of programming.

Seminar.

The class for 2D-vectors. The header file. The simplest functions of access and printing.

Lecture 6.

Constructors and destructors. The constructor by default. The copying constructor. Constructors as the type reduction operators.

Seminar.

Developing of class for the 2D-vectors: writing of constructors and destructors. Testing of the statical and dynamical creation of the objects.

Lecture 7.

Reloading of the operators and functions. Class friends.

Seminar.

Developing of the class for 2D-vectors: reloading operators of addition, subtraction and multiplication by a number.

Lecture 8.

Inheritance. Virtual functions. Abstract classes.

Seminar.

The class for 3D-vectors as an inherit of the class for 2D-vectors. Virtualization of the operation of printing output.

Lecture 9.

Inheritance and polymorphism. The basic idea. RTTI and dynamic_cast.

Seminar.

The program using an array of indexes to classes of 2D and 3D-vectors. Polymorphism on practice.

Lecture 10.

Exceptions. Exceptions and polymorphism.

Seminar.

A program using exceptions. Takeover of the exceptions of the operator "new".

Lecture 11.

C++-style of the input-output. Reloading of the operators of input-output. Managing of the format output. Manipulators.

Seminar.

A program of printing tables of sines and cosines using manipulators. The program of reading and recording to a file (C++ style).

Lecture 12.

Templates. The notion of generic programming and introduction to the standard library of templates (STL). Recommended literature on STL.

Seminar.

Programming of the simplest templates of functions. A program using std::vector.

Lectire 13.

Using STL: Containers, algorithms, iterators. Sequential containers vector and list.

Seminar.

The program of deleting from std::vector and std::list of all odd numbers.

Lecture 14.

Using STL: Associative containers "set" and "map".

Seminar.

A program using std::map.

Lecture 15.

Using STL: Algorithms. Function object.

Seminar.

A program using algorithms of sorting for std::vector. Sorting in increasing and decreasing order.

Lecture 16.

The standard library C++: string, complex. Useful functions. C++, what is left "behind the scene"?

Seminar.

A program using std::complex and std::string