# Fundamental types

* Fundamental types (bool, char, short, int, long, long long, float double, u.c.) - daži pārsteigumi, ja tos pārveido no viena tipa par otru, ja aizmirst inicializēt, ja pārsniedz CPU reģistrā glabājamo maksimālo vērtību.
* C++ has been specifically designed as “type unsafe” language. **Type casts:** Declaring a variable as a “bool” (instead of “int”) leads to mistakes
* Implicit type casts and uninitialized variables
* Literals – how to write integer, string, etc. literals (constants to initialize).

# Strings; I/O with “cin” and “cout”

* Patterns for input and output. Whitespace and other chars.
* Difference between LF, and CRLF line-breaks in Linux/Windows.
* Daži eksperimenti par ievadi un izvadi (pārsteigums, kas var rasties, ja pēc "string" mainīgā uzreiz ielasa "int" mainīgo - parādās neapēsts tukšums).

1. string s = "abc"; string t = "cde"; s += s + t[1] + s;

# Expressions

* Precedence and associativity
* Bit shift operators – bit operations preferred in various manipulations with bits/bytes.
* Operation values and their side-effects. Pre-increment and post-increment.
* Assignments; L-values and R-values
* Building an “abstract syntax tree”

**Restore parentheses:** y + 2 \* z ++ < 3 - w / 5.

**Short Function:** Write a short C++ function, isMultiple that returns true if and only if n is a multiple of m.

**Shifts etc.:** Write a short C++ function, isTwoPower, that takes an int i and returns true if and only if i is a power of 2. Do not use multiplication or division, however.

# Variables; Declarations and Definitions

* Declaring variables vs. Defining variables.
* Variable scope.
* Namespaces.
* What are header files
* Stand-alone functions vs. Class functions

1. Which of the following is not a valid C++ variable name? (Goodrich)

# If-else statements; switch statements

* You should be extra careful, when declaring a variable in a scope. (Re-declaring variable in a new scope means that it shadows the previous/earlier one.)
* Flowcharts for if-then and if-then else.

# Loops

* Can you modify the loop iterator inside a loop body? (Yes, but it is undesirable)
* What about for(;;) or for(int i = 0; ;) or … loops – with omitted versions?
* Can you use loop variable outside your loop? (Yes, if declared in outer scope)
* C++ does not really have a way to jump out of a nested loop – but you can use GOTO operator ☺
* Rewrite a nested loop with function calls.
* Rewrote a nested loop with Boolean flags.

Write a short C++ function that takes an integer n and returns the sum of all the odd integers smaller than n.

WriteashortC++functionthattakesapositivedoublevaluexandreturns the number of times we can divide x by 2 before we get a number less than 2.

# Arrays

* Arrays – defining arrays statically (if you know their size in advance)
* Arrays – defining arrays dynamically (if you do not know their size)
* Arrays – how they are initialized.
* Two-dimensional arrays – are they pointers to pointers? Or are they a big rectangular table?
* Pointer arithmetic.

1. Write a pseudo-code description of a method for finding the smallest and largest numbers in an array of integers and compare that to a C++ function that would do the same thing. (Goodrich)

# Few Remarks on Software Development

* No much teamwork in our class. (Our major lab - project #5 might involve some teamwork (everybody develops one class of C++ software), then they try to link this together.)
* Test Driven Development (<https://en.wikipedia.org/wiki/Test-driven_development> )  
  Kent Beck; Extreme Programming, Agile, etc.   
  They proscribe certain steps.
* Fakes/Mocks/Stubs… If you cannot “mock away” some behavior, the design is wrong.   
  (Complain – either to yourself or to the instructor, who forces to do such things).
* One GitHub repository – multiple directories.
* Clone on Windows.
* Clone on Linux (manual or done by Jenkins).
* Branches and tags will be introduced.
* Makefiles

# Linux Commands for Survival

mkdir ...   
ls   
ls -al  
ls -alt  
rm -fr <directory>  
cd

pwd ...

# Basic Git commands:

git checkout -b palindromes

git commit -m "Only palindromes"

git push origin palindromes

Branch specifier in Jenkins:

refs/heads/palindromes

git checkout master