Objects in C++
Exercises

Patrizia Scandurra

scandurra@dti.unimi.it

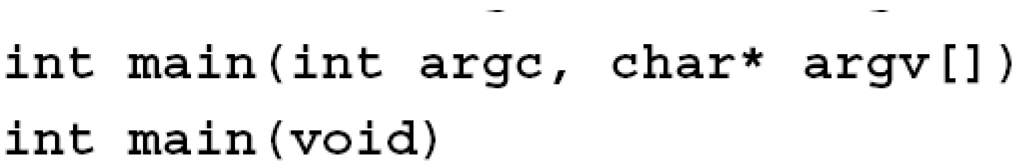
Info3

C++ source structure

- header files (.h, .hpp)
  - dependencies (`#include` directives)
  - type declarations
  - type definitions
  - function declarations
  - inline functions
- implementation files (.cpp, .cc)
  - dependencies (`#include` directives)
  - function definitions
  - `main()` function

main()

- each C++ program needs exactly one main() function
- program entry point
- possible signatures are



```
int main(int argc, char* argv[])
int main(void)
```

Exercises (1)

- Exercise I.1
  - learn how to use Dev-C++
  - write, compile and run a “Hello World!” console application
- Exercise I.2
  - create a console application which writes all its command line arguments to standard output (the screen), one per line
    - use index addressing
    - use a pointer directly

Exercises (2)

- Exercise I.3
  - create a console application which
    - takes exactly 5 command line arguments,
    - interprets them as integers,
      - hint: use int i = atoi(char* str) from stdlib.h
    - saves them into an integer array,
    - uses a bubble sort algorithm to sort the array and
    - writes the sorted array to standard output.

Exercises (3)

- Exercise I.4
  - write a class called `Complex`, which deals with complex numbers. Provide
    - convenient constructors
    - copy constructor
    - `operator=`
    - `operator+, operator-, operator*, operator/`
    - `operator+=, operator-=, operator*=, operator/=`
    - `re(), im(), norm(), phase()`
  - write a small test program

STL Exercises

1. **STL1**: declare a vector of integer values, stores five arbitrary values in the vector and then print the single vector elements to cout.
2. **STL2 :** declare a vector of string values, asks to the user to insert a sentence of one or more words, store each word in the vector and then print the sentence in reverse order
3. **STL3**: write STL2 with the iterator