Introduction

C++ in QF I - a course by Paweł Sakowski

Przemysław Kurek

Chair of Political Economy
Faculty of Economic Sciences
University of Warsaw

Class 00

C++ in QF I – plan of the course

Concentrated on general rules of object-oriented programming (OPP)

1. **basic C++ program**
   basic C++ syntax, compiling a C++ program, printing text, running a compiled program, pausing execution, understanding white space, adding comments to the source code, using an IDE
2. **simple variables and data types**
   declaring variables, assigning values to variables, printing variables, formatting numbers, understanding type conversion, introduction to characters, introduction to strings, introduction to constants,
3. **operators and control structures**
   arithmetic operators, if conditionals, using else and else if, the ternary operator, logical and comparison operators, increment and decrement operators, while loop, for loop
4. **input, output, and files**
   taking character input, discarding input, taking numeric input, taking string input, taking multiple inputs, reading in a whole line, validating input, creating file output, using file input
5. **defining your own functions**
   creating simple functions, creating functions that take arguments, setting default argument values, creating functions that return a value, overloading functions, understanding variable scope
6. **complex data types**
   working with arrays, working with pointers, structures, revisiting user-defined functions
7. **introducing objects**
   creating a simple class, adding methods to a class, creating and using object, defining constructors, defining destructors, the this pointer
8. **class inheritance**
   basic inheritance, inheriting constructors and destructors, access control, overriding methods, overloading methods, making friends
9. **namespaces & modularization**
   working with included files, the C preprocessor, understanding namespaces, linkage and scope
10. **working with templates**
    basic template syntax, creating inline templates, containers and algorithms

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

C++ in QF II – plan of the course

Concentrated on real applications in QF

1. **advanced opp**
   static attributes and methods, virtual methods, abstract methods, operator overloading, the << operator, multiple inheritance, virtual inheritance
2. **inheritance**
   class inheritance: basic inheritance, inheriting constructors and destructors, access control, overriding and overloading methods, virtual functions, passing of arguments, using overloading operators, private and public data, defining constructors and destructors,
3. **error handling and debugging**
   debugging techniques, returning error codes, using `assert()`, catching exceptions
4. **dynamic memory management**
   static and dynamic memory, allocating objects, allocating arrays of dynamic size, returning memory from a function or method, the copy constructor and the assignment operator, static object type casts, performing dynamic object type casts, avoiding memory leaks
5. **more advanced methods of OPP**
   virtual constructors and bridge pattern, separating interface and implementation, more complicated design patterns, using of templates, advanced OOP: `static/virtual/abstract` methods, multiple and virtual inheritance,
6. **random number class**
   developing random number class with reusable interface and adequate random number generator, implementation of antithetic sampling,
7. **pricing of exotic derivatives**
   Monte-Carlo for path dependent exotic derivatives, template pattern, pricing of Asian options,
8. **interfacing `C++` and Excel**
   the object model in Excel, accessing Excel objects from `C++`, getting data into `C++` from Excel, display vector and matrix data in Excel, displaying functions in Excel
9. **integration of `C++` with `R`**
   `Rcpp` package, `inline` package, core data types, plotting from `C++` via `R`, `Rcpparmadillo`,

Grading

- Final exam 50 pts.
- Home taken project 50 pts.
- 100 pts. to collect in total
- Minimum requirement to pass:
  1. at least 60 pts. in total,
  2. and at least 25 pts. from the final exam,
  3. and at least 25 pts. from the project
- Online attendance is not mandatory, however strongly advised.
- Minimum threshold will be lowered to 50 pts. for active students.
- Also, some special events may happen, which will allow you to get some extra points or lower your threshold.

Contact

- Moodle! All questions that can be solved publicly will be redirected to ask in public on Moodle.
- Office hours: Tuesdays 16:45-17:45, MS Teams. Please, inform me in advance if you want to come!
- Email (in cases where a public question can not be asked): `pkurek@wne.uw.edu.pl`

How to learn?

You need **a lot** of **regular** practice! We should no ask if you can learn C++, but only when.

- **Try online challenges:**
  `https://play.google.com/store/search?q=sololearn` +
  `https://www.bluestacks.com/`
- **Find another online course:**
  `https://www.sololearn.com/`
  `https://www.coursera.org/`
- **Online all-in-one knowledge base:**
  `http://www.cplusplus.com/`
- **Books:** `http://stackoverflow.com/questions/388242/the-definitive-c-book-guide-and-list`
- **Search the web by yourself as soon as poosible!**
- **Redo classes, avoid copypasting code, do exercises!**

Programming environment

We need 'a place' where we will write our programs:

- We will start with `https://www.onlinegdb.com/` to make the start extremely easy for you. However, it is a toy. For sure the time will come to use something more serious:
- If you do not want to use computer in your future, or for some unknown reason you love Windows you can start here: `https://code.visualstudio.com/docs/languages/cpp`
- If you want to use computer in your future you should start using Unix Terminal (in Linux or Mac). Ask me for directions!
- You are allowed to use anything, that works. Try different solutions: IDE's, text editors, compilers...

Uniwersytet Warszawski
Wydział Nauk
Ekonomicznych