Exercises HowTo

Sébastien Ponce sebastien.ponce@cern.ch

CERN

October 2020

Setup requirements

Required

- decent C++editor
- any C++compiler supporting C++17

Good to have

- `git` for getting/managing your code
- `gdb` to debug your problems
- valgrind, kcachegrind, cppcheck for corresponding exercises

C++and python specific needs

- python3, libpython3-dev
- ctypes, matplotlib, numpy python packages

How to get exercises' code

Best way

```
git clone https://gitlab.cern.ch/sponce/cpluspluscourse
cd cpluspluscourse/code
```

Alternative

```
wget http://cern.ch/sponce/C++Course/code.tgz
tar xzf code.tgz
cd code
```

Anatomy of an exercise

Each exercise is in a subdirectory with mainly 3 sets of files

*.hpp and *.cpp files

- the code to understand and fix/complete
- you never start from scratch

Makefile / cmake

- prepared Makefile for easy compilation
- "make" and "make clean" are available
- cmake is also supported

solution subdirectory

- the solution to the exercise ("make solution")
- please do not use before trying !

Just follow the instructions

- Each exercise comes with a set of instructions in the course
- See for example exercise on polymorphism
- Also present in exercise subdir in markdown format

Exercise Time

- go to code/polymorphism
- look at the code
- open test.cpp
- create a Pentagon, call its perimeter method
- create an Hexagon, call its perimeter method
- create an Hexagon, call its parent's perimeter method
- retry with virtual methods

How to test your setup

Just use hello exercise

- go to code/hello
- follow the README

Practically

```
cd code/hello
```

- g++ and make

```
make;
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:.
./hello
```

- valgrind, callgrind and X11

```
valgrind --tool=callgrind ./hello; kcachegrind
```

- cppcheck

```
cppcheck .
```

How to get help

Several channels available

- Training Notebook
- Mattermost channel
- Zoom rooms with mentors

All links and details on the indico page of the event