SUPA COO (C++)
Lecture 4 – 12 Nov 2015

Dr. Adrian Buzatu
University of Glasgow, adrian.buzatu@glasgow.ac.uk

Thanks to to some usage of previous lecturer's materials
(S. Allwood-Spiers & W. H. Bell)

Classes

Classes are what makes C++ powerful.
Classes are new types we introduce (like int, double).
We solve complex problems.
We define operators on them, as +, -, or <
We define functions that can work on them, called methods.
They can be templated too.

Vector in 2D

Properties: x of type double; y of type double; size of type double and value sqrt(x*x+y*y)

What it knows to do:

```
//constructor
Vector v(3.0,4.0);
//getters
std::cout<<“x=“<<v.GetX()<<“ y=“<<v.GetY()<<std::endl;
std::cout<<“size=“<<v.GetSize()<<std::endl;
//setters
v.SetX(10.0);
v.SetY(2.0);
//do things
v.Print();
v.DoubleTheValues();
```

Vector in 2D

We can define operators on it

```
//operator overloading
v+=Vector(1.0,2.0)
v*=2.0;
Vector v1(2.0,2.0);
Vector v2(1.0,1.0);
v1+=v2;
Vector v3=v1+v2;
std::cout<<“are two vectors identical “<<v1==v2<<std::endl;
std::cout<<“is v1 smaller than v2“<< v1<v2 <<std::endl;
std::cout<<“dot product v1*v2=“<<v1*v2<<std::endl;
```

How do we compile the class

How to build the class code as shared library
in TutorialCPP/Vector
How to use the class code in executable
in TutorialCPP/testVector

You can do the same for a 3D vector instead of a 2D vector
How to build the class code in TutorialCPP/Vector
How to use the class code in TutorialCPP/testVector

How do we compile a package

The Makefile from TutorialCPP is very complex, as it does many things: compiles for both Linux and Mac, for executable and shared library.

TutorialCPP/Makefile
TutorialCPP/Makefile.arch
TutorialCPP/Makefile.common1
TutorialCPP/Makefile.common2
These are common code, allowing in each folder to have only few things dependent on the folder:

include ../Makefile.arch
include ../Makefile.common1
COMPILE_AS=exe
WHICH_OTHER_SHARED_LIBRARIES_ARE_NEEDED_TO_COMPILE=-L$(PATH_TO_SHARED_LIBRARIES_FROM_THIS_PACKAGE) # ex: add -lBlah for Blah library
include ../Makefile.common2

Standalone class and compilation – Lab3

Let's solve the lab2 problem with a class called Planet, called Lab3. I posted online such a solution.

Please go through them to learn

- how to create and use a class,
- but also how to compile it both simply, and with Make, and as a shared library.

Four folders, each with HOW_TO_RUN.txt.

We will go through them now, please see all of them by yourself.

We are then prepared to tackle complex projects.

Lab4 – simulating the solar system

The task for Lab4 next Tuesday is already online.

Please go first thoroughly through the solution for Lab3 and give it a try to Lab4 beforehand, so that we can use this last lab to ask questions, to make sure we finish it in time.

It is not complicated, but a very nice application to see how useful C++ is.