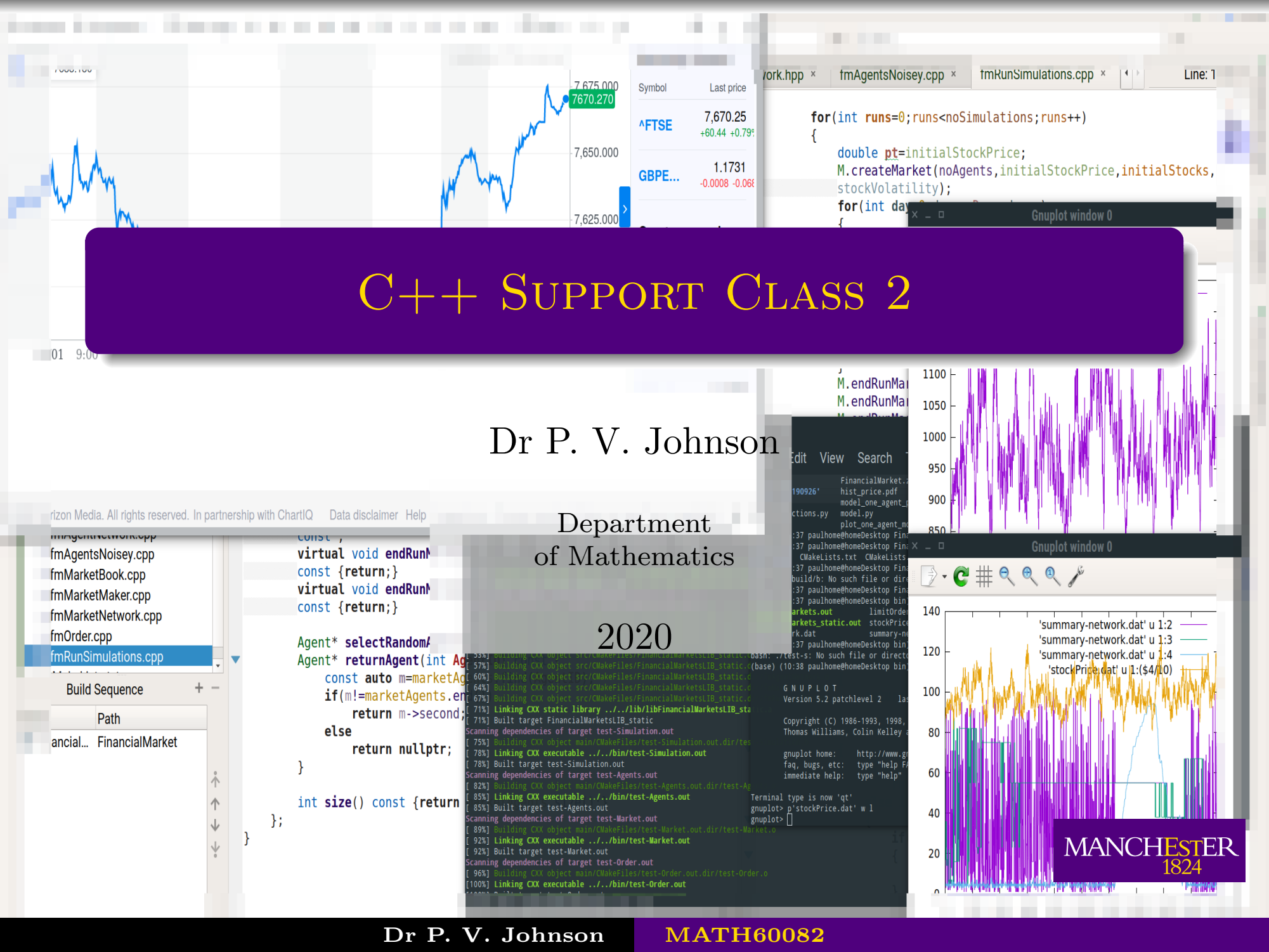
C++ Support Class 2

Dr P. V. Johnson

Department
of Mathematics

2020



Last Time...

- Topics:
  - Computers and Programs;
  - Syntax of C++;
  - Data and Variables;
  - Input and Output.
- Aims:
  - Understand the idea of programming a computer;
  - Write a simple program to input and output data.

MANCHESTER 1824

Today...

- Topics:
  - Flow control - if, else, for, do
  - Functions - how and where to use them
  - Pointers - dynamic allocation, pass by reference
- Aims:
  - Use functions, loops and if statement to control a program.
  - Understand the concept of a pointer.

MANCHESTER 1824

If, Else If and Else

- We can use `if`, `else if`, and `else` to control flow through the program.

```
int i;
cout << " Enter a number " << endl;
cin >> i;
if(i<0)cout << " i is negative" << endl;
else if(i==0)cout << " i is zero" << endl;
else cout << " i is positive" << endl;
```

If, Else If and Else

- To execute more than one command on an `if` condition use blocks

```
if(condition){
// lots of commands in here
}
else {
// and in here too.
}
```

MANCHESTER 1824

Tasks

- Write a function that takes the arguments a, b and c, computes and then displays the roots of the quadratic equation

$$ax^2 + bx + c = 0$$

You will need to identify whether the roots are real or complex. If the roots are complex, display the results in the form $A + Bi$.

Getting lost? Click here for the solution to this example, pg 9–12.

- The general form for a loop is

```
for(initialisation; condition; increment)
statement;
```

- The general form for a loop is

```
for(initialisation; condition; increment)
statement;
```

- We can loop over multiple commands using a block

```
for(int i=0;i<10;i++){
temp = i*10;
cout << " value " << temp << endl;
}
```

Exiting a loop

- The command `break` can be used to exit a loop.

```
for(int loop=0;loop<iter_max;loop++)
{
solve_for_U(u,y,U);
if(residual(x,y,U)<tolerance)break;
}
```

The while loop

- Another alternative loop iteration is the `while` loop.
- Their functional form is:

```
while(condition)statement;
```

- The statement is executed until the condition is true.
- The condition is evaluated **before** the statement.

```
int i=0;
while(i<100)
{
i++;
std::cout << " i= " << i << std::endl;
}
```

The do-while loop

- Another alternative loop iteration is the `do-while` loop.
- Their functional form is:

```
do {statement;} while(condition);
```

- The statement is executed until the condition is true.
- The condition is evaluated **after** the statement.

```
do {
solve_for_U(u,y,U);
}
while(residual(x,y,U)<tolerance);
```

MANCHESTER 1824

Tasks

- Write a loop to calculate and output the first n fibonacci numbers.
- Write a loop to calculate and output the binomial coefficients for $0 \leq k \leq n$:

$$\binom{n}{k}$$

- The general syntax for a function is:

```
data type function_name(arguments)
{ function statements }
```

- Functions must be declared before the main program.
- All functions must return a value of the data type specified in the declaration.
- Even if this is `void`!

Example Function

```
#include<iostream>
using namespace std
// square an integer
int square(int i)
{ return i*i; }
// Main Program
main(){
int number=5
cout << square(number) << endl;
}
```

Accessing the math library

- Simply include the library at the top of your code:

```
#include<cmath>
```

- All of the trigonometric, hyperbolic and exponential functions are present.
- There is also a `pow(x,y)` to raise `x` to the power `y`.
- and a sqrt() function.

MANCHESTER 1824

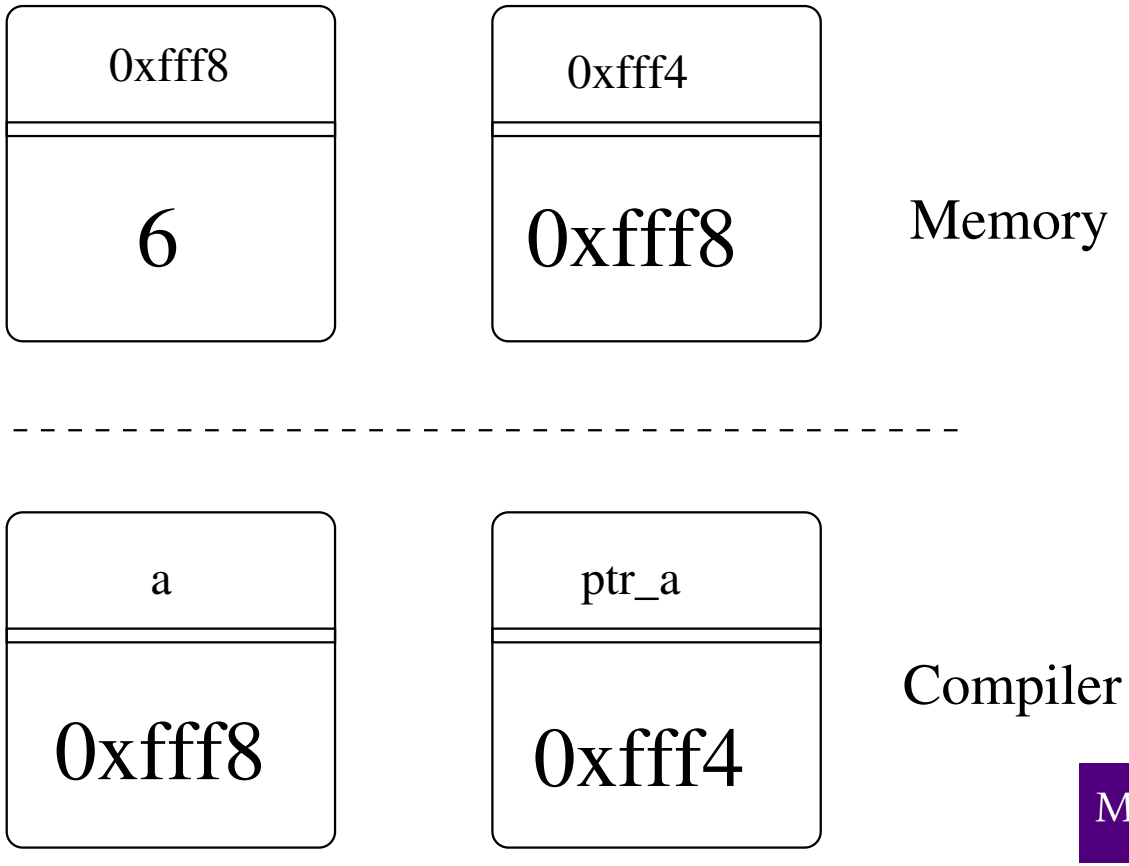
- A function must be defined before it can be called.
- Use prototypes to declare functions before they are used.

```
data type function_name(arguments)
```

- The main body of the function can be placed somewhere else in the code (or even a separate file)

What's in a name?

- Pointers are an important mechanism in any computer program.
- In many languages pointers are *hidden* from the user.
- Pointers store the **location** of a value, rather than the value.



- When we pass a variable into a function, the stored **value** is copied into the function, not the variable itself.
- To change the value of the variable itself, we must pass a reference memory location of the variable into the function.

```
void swap(double &a,double &b)
{   /*  stuff in here */  }
main()
{
double a=1,b=2;
swap(a,b);
cout << " a " << a << " b " << b;
}
```

YOU DON'T NEED TO KNOW ANYTHING ABOUT POINTERS TO CODE MATHEMATICAL FUNCTIONS

MANCHESTER 1824

YOU DON'T NEED TO KNOW ANYTHING ABOUT POINTERS TO CODE MATHEMATICAL FUNCTIONS

JUST REMEMBER TO PUT & WHEN CHANGING VALUE INSIDE A FUNCTION!!!!

MANCHESTER 1824

New Project – or New Function?

- Functions can be used to structure your code.
- Imagine you have solved a problem with some code in the main function.
- But now you want to solve a different problem.

- Do I need to open a new project? You CAN'T make another main function.

MANCHESTER 1824

New Project – or New Function?

- Functions can be used to structure your code.
- Imagine you have solved a problem with some code in the main function.
- Your code look like this:-

```
int main()
{
/* ...  SOME SOLUTION IN HERE ...  */
}
```

New Project – or New Function?

- Functions can be used to structure your code.
- Move your code out into a void function (no parameters) and you can do something else in the main function.

```
void mySolution()
{
/* ...  SOME SOLUTION IN HERE ...  */
}

int main()
{
/* ...  DOING SOMETHING ELSE ...  */
}
```

New Project – or New Function?

- Functions can be used to structure your code.
- Move your code out into a void function (no parameters) and you can do something else in the main function.

```
void mySolution()
{
/* ...  SOME SOLUTION IN HERE ...  */
}

int main()
{
mySolution();   /*  runs solution  */
}
```

- BUT still be able to run that code if required.

MANCHESTER 1824

Tasks

- Try writing a different function to solve each of the problems you have encountered so far.
- You are given that

$$\phi(\eta, \xi) = e^{-\lambda_b \eta} e^{-\lambda_s \xi}$$

and

$$\Gamma(x, t) = (1 - \phi(x^2 t, \sqrt{t})) + \sin(\phi(t^{-1}, x t^{-2})).$$

Write two functions to evaluate these quantities. Calculate Γ when $\lambda_b = 1.3$, $\lambda_s = 0.26$, $x = \pi^2$ and $t = \sqrt{3}$.

Think about how you can include parameters in your functions.

File Input and Output

- To read and write to files we must include the `fstream` library.
- Input streams have type `ifstream`, and output streams `ofstream`

```
ifstream file_input; // an input file stream
ofstream file_output; // an output file stream
```

- `ifstream` and `ofstream` have intrinsic functions to open and close files.
- We can also check if the file is open with the `is_open()` function.

```
file_input.open("input.in"); // open file input.in
if(file_input.is_open()) // check file is open
```

Write to file

Tasks

- Follow the solutions (click here) to set up a program to write to a file on your system.
- Write down the first two terms in the Maclaurin series for $\exp(x)$.
- Write a program to put data into the file `data.csv`. You must generate four columns of data to enter into the file: the value of x, the numeric solution for $\exp(x)$, the two term Maclaurin series approximation to $\exp(x)$, and the difference between the numeric solution and the approximation. Generate data for at least 50 values of x in the range $x \in [-0.5, 0.5]$. Use a comma to separate columns, you will need to use a for loop to create rows of data.

Summary

- Topics:
  - Flow control - if, else, for, do
  - Functions - how and where to use them
  - Pointers - pass by reference
- Aims:
  - Use functions, loops and if statement to control a program.
  - Use pass by reference.

MANCHESTER 1824