University of Hull

Writing Programs

C# Programming

Being a Computer Programmer

- Nowadays it is almost socially acceptable to admit that you can write programs
  - Although you might get asked to fix a lot of computers if you do
  - …and people will wonder why you aren't rich
- I tell people I am a "Software Engineer"
- This sounds more classy, and is closer to what I actually do

What is Programming?

- Programming:

  "Deriving and expressing a solution to a problem in a form which a computer can understand and execute"

- You have to know how to solve the problem **before** you write the program
- The computer has to understand your instructions

From Problem to Program

- The starting point for a program is a problem
- Throughout this course we are going to work on the basis that we are solving problems for customers
- This helps us focus on the job in hand:
  - If we don't make the customer happy, we don't get paid…

Working with Customers

- If you can learn how to work with customers you will become a great programmer
- Computer Scientists are not noted for their "people skills" (not sure why) but if you can get good at this angle your future in the business is assured

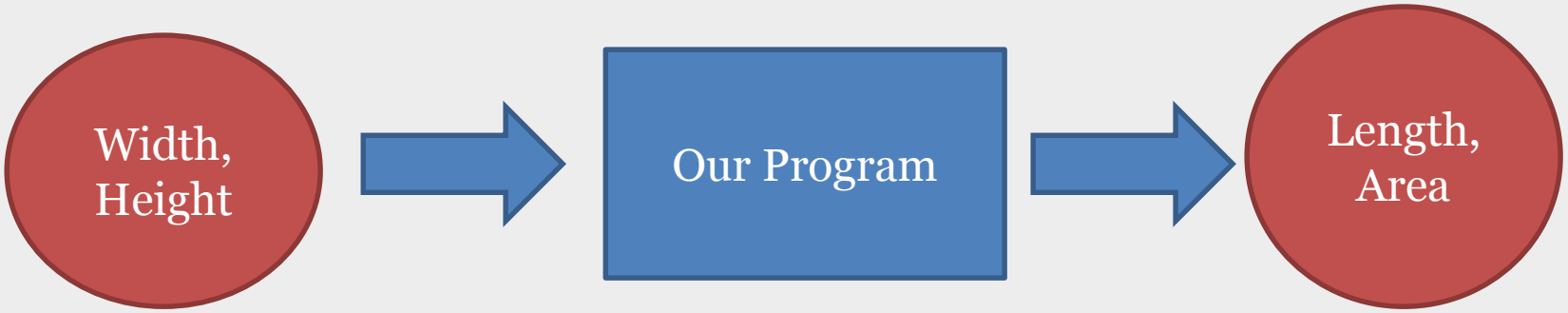
Solving the Wrong Problem

- This is one of the most common reasons why software projects fail
- The programmers created a solution to the wrong problem
- There are lots of reasons why this can happen
- Not all of these reasons are technical

Double Glazing Problem

- Given the width and height of a window we need to calculate:
  - The amount of wood needed for the frame
  - The area of the glass required

Data Processing



- We can view our program as a data processor
- We can specify what goes in and what comes out

Adding Metadata

- Metadata is **very** important
  - there is always a question in our exam about metadata
- Metadata for us is *"data about data"*
- Meta in this context means "stepping back" from the problem in order to consider the meaning of the things in it

The Importance of Metadata

- The metadata is how we get from data to information
- Saying a window is “3” wide does not tell us anything, we also need to know that window width is measured in metres
- To create a useful solution we need to gather all the metadata about it

Spot the Metadata:

- Some of these statements are metadata:
  - Windows sizes are measured in metres
  - Wood is ordered in feet lengths
  - The lounge window is 2 metres wide
  - The largest piece of wood you can have is 5 metres long
  - Window frames are 2 inches thick

Spot the Metadata:

- Some of these statements are metadata:
  - Windows sizes are measured in metres
  - Wood is ordered in feet lengths
  - The lounge window is 2 metres wide
  - The largest piece of wood you can have is 5 metres long
  - Window frames are 2 inches thick

Defining Inputs

- Width:
  - Measured in metres
  - Smallest value 0.5 largest 3.5
- Height
  - Measured in metres
  - Smallest value 0.5 largest 2.0

Defining Outputs

- Length of wood
  - Measured in feet (conversion of 3.25 feet per metre
- Size of glass
  - Measured in square metres

Proving it Works

- The final thing we need is a test to prove that it works:

  "If I give the program the inputs 2 metres high and 1 metre wide the program should print out: 4 square metres of glass and 19.5 feet of wood"

- If the program does this the customer must pay me for my work

Getting it in Writing

- It is very important if you do a job for money that you make sure that:
  - You get paid
  - You don't end up doing too much work
  - You don't get sued by your customer
- Getting your specification signed off is the key to achieving these things

Customer Involvement

- The worst thing you can do at this point is go off and build the solution
  - The customer should continue to be involved in the development
- Often the customer will have strong opinions on how the program is to be used
- They also might change their requirements when they see what you have done

Writing the Program

- If you don't know how to solve the problem yourself you can't write the program
  - A program is just your explanation of the how to solve the problem
- Working out how you would perform the solution is a very good starting point for writing the program

Programming Languages

- We can't just use English to tell a computer what to do
  - Computer are too stupid to understand it.
  - English is much to vague.
- Instead we have to use special, unambiguous, languages which can be converted into computer instructions

Different Languages

- There are hundreds of different programming languages
  - each has its own fans and detractors
- They all do the same job though
  - provide a way to tell a computer what to do
- As a programmer you will need to be able to use several languages and will have to learn new ones

The C# Language

- C# is based on the Java language, which is based on the C++ language, which is based on the C programming language
- Once you have learnt C# you should be able to pick the others up quite easily
  - Although there are some differences that you need to know

Summary

- Programs tell a computer what to do
- Information that adds context to your data is called “metadata” (data about data)
- Programming is also about finding out what the customer wants you to do
- The program you write is expressed in a special programming language