

technocamps

Inspiring | Creative | Fun
Ysbrydoledig | Creadigol | Hwyl



Swansea University
Prifysgol Abertawe



PRIFYSGOL
ABERYSTWYTH
UNIVERSITY



PRIFYSGOL
BANGOR
UNIVERSITY

University of
South Wales
Prifysgol
De Cymru



ESF



Llywodraeth Cymru
Welsh Government

Ewrop & Chymru: Buddsoddi yn eich dyfodol
Cronfa Gymdeithasol Ewrop

Europe & Wales: Investing in your future
European Social Fund

ALICE:

An introduction to
programming



Computer Science

What is Computer Science?

- Do you know the **difference** between ICT and Computer Science?
- Any suggestions as to what jobs you could do if you were a Computer Scientist?

Programming

What is programming?

- Who here has experience with programming?

This are the software we will be using today:

Has anybody used or seen these before?



Computers are SILLY!

Why on earth am I calling computers silly?

I am SILLY!



Remember: Computers do not ask questions!

Let's think about how we could get a computer to make a cup of tea...

Importance of clear instructions

How could we command a computer to make us a cup of tea, step-by-step?

Providing clear, understandable instructions is very important to program. The order of instructions is also very important.



Let's test this theory...

The task:

I will be the programmer. You will each be one of my computers.

I will deliver a range of instructions. Your task is to interpret them and react.

You will each need a piece of A4 paper and a pen to hand!



What do we need our program to do?

Now that we know the basic skills required to program, what exactly happens to get the program to perform your instructions?

WRITE

COMPILE

RUN

DEBUG



These are the processes needed to be completed to program :

'Writing' the program

The writing of the code is the actual typing out of the instructions bit-by-bit for the computer to complete.

```
class MyFirstJavaProgram{
    public static void main(String [] args){
        System.out.println("Hello World!");
    }
}
```

Above is an example of code for a computer in a programming language called Java. (It looks a bit scary at the moment...)



‘Compiling’ the program

The next step is to ‘compile’ your code.

- Simply writing the code won’t do anything.
- Compiling the code will turn it into a language that the computer understands.

Here is an example of what some binary code may look like:

0100110101010010101010101011010100101110101101110101010101

But, speaking to the computer in this language is very difficult. Let’s stick to the ‘human friendly’ languages!



‘Running’ the program

You will not know your instructions/code works unless you run your program.

- What if it doesn't run how we expected it to?
- It is normal if it doesn't work 1st time.
Or even the 101st time!
- Even the experts make mistakes.
- We need to ‘debug’ our code to make it better and run the way we command it to.

But, what is debugging?



'Debugging' the program

Debugging is a way for the computer to find any errors within your code and alert you of them. That way you can fix them.

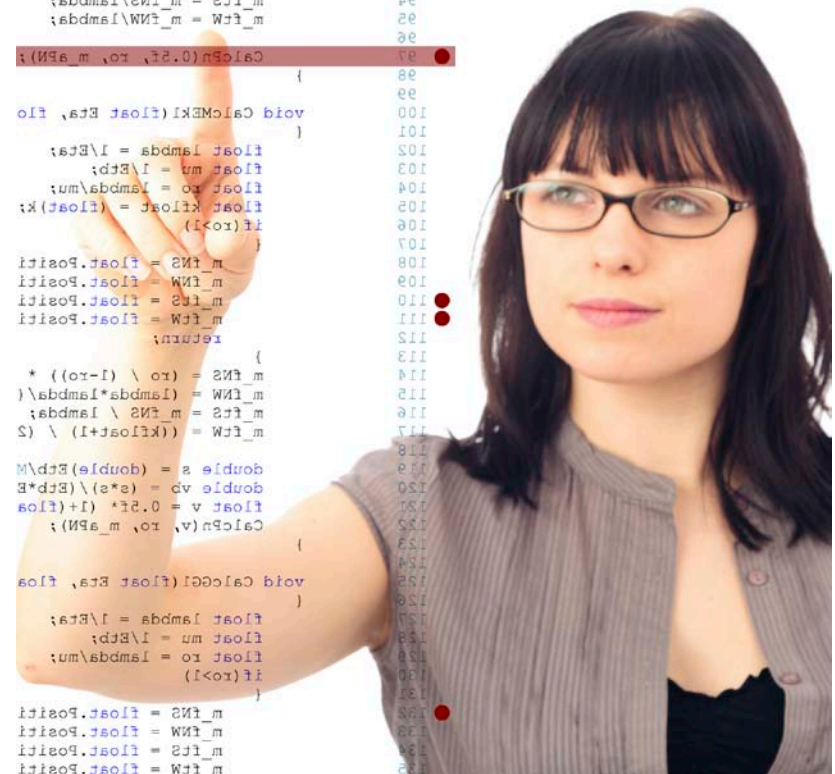
It is thought:

It is called 'Debugging' because a lady named Grace Hopper in the 1940s discovered the reason her computer was no longer working properly was due to a dead moth she found inside the computer...

```

87 m_fNW = float.Position
88 m_fTS = float.Position
89 m_fTW = float.Position
90 return;
91 }
92 m_fNS = (ro \ (1-ro))*I
93 m_fNW = ro*ro \ (2*(1-ro
94 m_fTS = m_fNS\lambda;
95 m_fTW = m_fNW\lambda;
96
97 CalcPn(0.5f, ro, m_fNS)
98 }
99
100 void CalcMEKI(float Ets, floa
101 {
102 float lambda = I\Ets;
103 float mu = I\Efb;
104 float ro = lambda\mu;
105 float Kfloat = (float)K;
106 if(ro>1)
107 {
108 m_fNS = float.Position
109 m_fNW = float.Position
110 m_fTS = float.Position
111 m_fTW = float.Position
112 return;
113 }
114 m_fNS = (ro \ (1-ro)) *
115 m_fNW = (lambda*lambda)\
116 m_fTS = m_fNS \ lambda;
117 m_fTW = ((Kfloat+1) \ (2
118
119 double s = (double)Efb\M
120 double vp = (s*s)\(Efb*E
121 float v = 0.5f*(1+(float)
122 CalcPn(v, ro, m_fNS);
123 }
124
125 void CalcG1(float Ets, floa
126 {
127 float lambda = I\Ets;
128 float mu = I\Efb;
129 float ro = lambda\mu;
130 if(ro>1)
131 {
132 m_fNS = float.Position
133 m_fNW = float.Position
134 m_fTS = float.Position
135 m_fTW = float.Position

```



Welcome to ALICE!

Let's go over an example of how we make soup:

Soup

Ingredients

Actions

This software is called a “Drag and Drop” type environment to learn the basics of programming.

You don't have to worry about spelling errors and general syntax errors, all of the instructions are provided.

All you have to do is make sure you are using the right instruction correctly and in the right order.

Conditional statements & iteration

```
IF *condition occurs* THEN  
do *list of instructions*
```

```
ELSE
```

```
*alternative list of instructions*
```

```
DO *list of instructions*  
UNTIL *condition occurs*
```

```
FOR *number of times*  
DO *list of instructions*
```

Pseudocode

...Sometimes even the most experienced programmer will need to sit down and write out his/her code in this manner below:

Pseudocode =

Output type string "Hello World"

Java code =

```
class MyFirstJavaProgram{  
    public static void main(String [] args){  
        System.out.println("Hello World!");  
    }  
}
```

Remember...

It is OKAY to make errors!

Even the most experienced programmers forget the odd “;” or “}” here and there.

Our task:

Let's firstly familiarize ourselves with ALICE and work our way through the 4 tutorials available.

Then, let's get stuck in and make a new project.

Don't forget to consider some of the following:

- Create a simple project for an object to move around the world.
- Experiment further by adding sounds and have objects interact.
- Make a movie...

