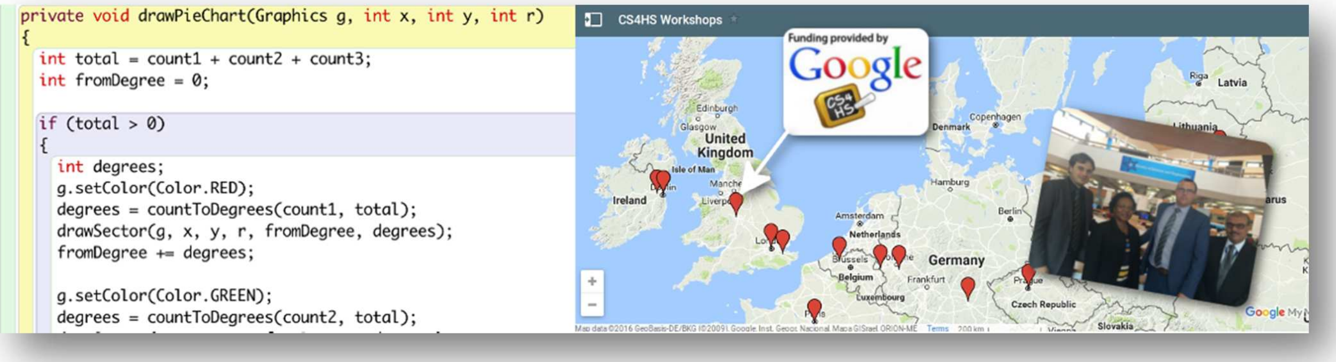
UNIVERSITY OF WOLVERHAMPTON
KNOWLEDGE ▪ INNOVATION ▪ ENTERPRISE



```
private void drawPieChart(Graphics g, int x, int y, int r)
{
  int total = count1 + count2 + count3;
  int fromDegree = 0;

  if (total > 0)
  {
    int degrees;
    g.setColor(Color.RED);
    degrees = countToDegrees(count1, total);
    drawSector(g, x, y, r, fromDegree, degrees);
    fromDegree += degrees;

    g.setColor(Color.GREEN);
    degrees = countToDegrees(count2, total);
```

Java Programming

for

GCSE & A-Level

Funding provided by Google CS4HS

Dr. Patricia Davies,   Dr. Liam Naughton,   Dr. Fernando Loizides

A free MOOC offered by the University of Wolverhampton

Introduction

The University of Wolverhampton is one of **only three** U.K. institutions to be awarded funding under the **Google “Computer Science for High School”** initiative this year. Academics from the School of Mathematics & Computer Science have been awarded funding to develop a free online course to provide continuing professional development for secondary school teachers.

This free **MOOC** (Massive Open Online Course) is designed to prepare secondary teachers of Computer Science to teach the programing elements of the new **GCSE** and **A Level** syllabuses. You will learn the basics of **computer programming**, **algorithms** and **object-oriented programming** through the Java programming language.

The course includes sections to assist you with underlying **mathematics** concepts, and **robotics applications** using **The Finch**, a small robot designed to inspire and engage learners in programming. The Finch was developed at the Carnegie Mellon University CREATE Lab. The research projects that led to the Finch used "participatory design", which means that they deeply involved both students and teachers at all stages of the design process. The Finch is designed to inspire and delight students learning computer science by providing them with a tangible and physical representation of their code.

The MOOC will provide **comprehensive course materials** and plenty of **programming exercises**, which could be used in classes you teach. There are no deadlines and the course is self-paced.

A **Free Finch Robot** will be made available to each local school, provided that participating teachers complete all the MOOC assignments no later than **Sunday 29th January 2017**.

Course Instructors

Dr. Patricia Davies

Patricia was born in Sierra Leone. In 2013 she completed a Doctorate in Education in the area of Educational Technology Leadership and Policy, at the University of Manchester. From 2002 to 2014 she was head of high school Computer Science at ACS Cobham International School in Surrey, UK, where she taught Computer Science in the Advanced Placement and International Baccalaureate programmes. Patricia holds a Bachelor's degree in Computer Science and Mathematics from Spelman College, and two Master's degrees—one in Mathematics from the University of California at Berkeley, and the other in Educational Technology from Teachers College, Columbia University in New York. Her research interests include educational data mining and learning analytics.

Dr. Liam Naughton

Liam hails from Ireland. He received his Ph.D. in Mathematics from NUI, Galway in 2010. He obtained his M.Sc. in Mathematics in 2006 and his B.Sc. in Computing Studies & Mathematical Sciences in 2005. After completing his Ph.D. he spent several years as the Coordinator of the Mathematics Learning Support centre at NUI, Galway before taking up a lectureship in Mathematics at the University of Wolverhampton in 2015. His research interests lie in computational group theory. He has given presentations at many international conferences and he is a Fellow of the Higher Education Academy.

Dr. Fernando Loizides

Fernando was born in Cyprus. He received his Ph.D. from City University London in 2012. He is a Lecturer in Computer Science at the School of Mathematics and Computer Science at the University of Wolverhampton. He also holds an honorary Fellowship at the Cyprus Interaction lab, Cyprus University of Technology. His main area of research lies in Information Interaction, HCI and Digital Libraries, focusing on Information Seeking, Information Architecture and User Experience, with a special interest on user interfaces. He has extensive experience in user study design and facilitation using cutting edge technologies, eliciting user requirements and performing systems evaluation, both within industry and academia.

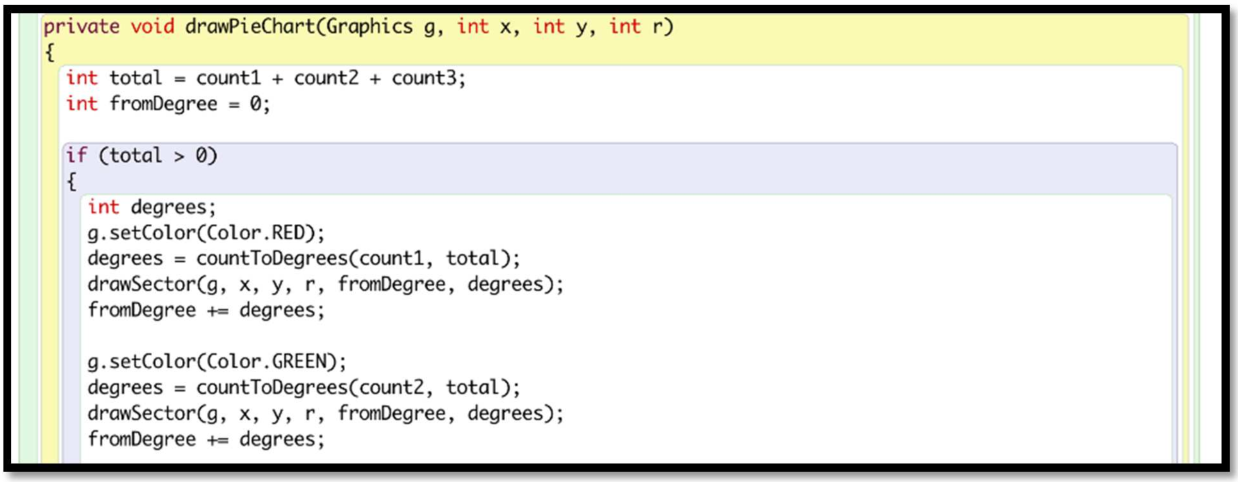
Course Details

The MOOC will be delivered using the **Canvas Network** platform and will be officially launched during the **Wolverhampton Festival of Computer Science** on **Saturday 15th October 2016**. The MOOC commences proper on **Monday 31st October 2016** and runs for eight weeks.

Each taught week will consist of three overlapping sessions

- **Java Programming** – Dr. Patricia Davies.

Each programing session will be broken up into **short topics**. Each of these short topics will be accompanied by a **3-5 minute video lecture** followed by an online multiple choice quiz.



```
private void drawPieChart(Graphics g, int x, int y, int r)
{
  int total = count1 + count2 + count3;
  int fromDegree = 0;

  if (total > 0)
  {
    int degrees;
    g.setColor(Color.RED);
    degrees = countToDegrees(count1, total);
    drawSector(g, x, y, r, fromDegree, degrees);
    fromDegree += degrees;

    g.setColor(Color.GREEN);
    degrees = countToDegrees(count2, total);
    drawSector(g, x, y, r, fromDegree, degrees);
    fromDegree += degrees;
```

Fig 1 Java program to construct a pie chart.

- **Programming the Finch Robot** – Dr. Fernando Loizides.

The material discussed in the programming sessions will be brought to life using the **Finch**. Short videos will be provided to help you to visualise the Finch in action. You are free to use these videos in your own teaching if you so desire.

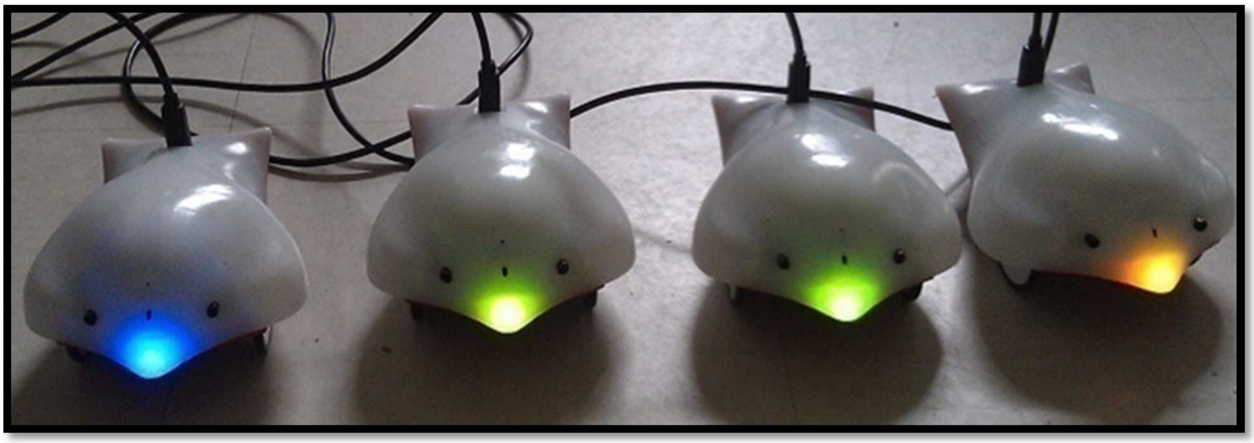


Fig 2 The Finch Robot.

- **Mathematics for Programming** – Dr. Liam Naughton.

Some of the mathematics underpinning the programming topics will be discussed in these sessions. Like the programming sessions these will be broken up into short topics. Each short topic will be accompanied by a **3-5 minute video** lecture followed by an online multiple choice quiz.

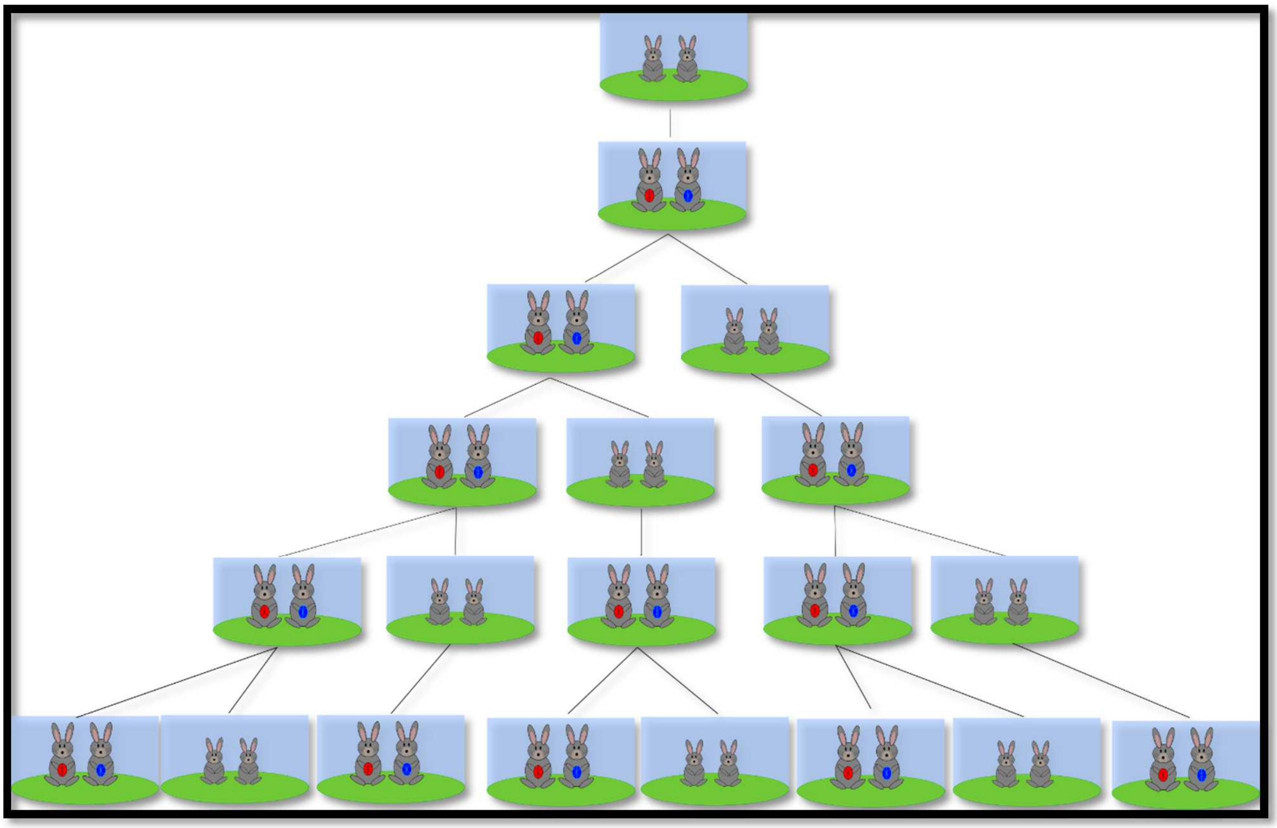


Fig 3 Fibonacci numbers illustrated by pairs of reproducing rabbits.

Each teaching session will also include a **downloadable comprehensive handout** covering the topics in more detail. Video lectures will also include a **downloadable transcript**. You are **free to use** these handouts and transcripts in your own teaching if you so desire. A more detailed description of the syllabus and a description of the correspondence between the syllabus and the GCSE and A-level syllabuses is given in the Appendix.

Wolverhampton Festival of Computer Science

The MOOC will be officially launched at the

Wolverhampton Festival of Computer Science

which takes place on

Saturday 15th October 2016

from 10:00 to 16:00 at City Campus, Wolverhampton. **This event is free but attendance is capped at 150.** You can register today by visiting

http://www.eventbrite.co.uk/e/festival-of-computer-science-tickets-27192343030

All participants who register before the festival will be entered into a draw to win one of fifty Amazon vouchers worth £50. The festival programme is shown below.

10:00 – 10:55	Arrival/Registration and **Exhibition**
11:00 – 11:30	Welcome Address
11:35 – 12:00	Introduction to **Emerging Technologies** at UoW
12:05 – 12:30	Intro to the MOOC: “Java Programming for GCSE & A-Level”
12:35 – 13:10	Lunch and **Exhibition**
13:15 – 14:45	Parallel Sessions (I, II, III, 25 mins sessions)
I.	Mathematics for Programming
II.	Programming with The Finch
III.	Programming in the new GCSE and A Level Computer Science Curriculum
14:50 – 15:30	Presentation from Google
15:35 – 16:00	Closing Q&A

Contact

If you would like any more information on the MOOC or the Computer Science Festival then please get in touch with

Dr. Patricia Davies	**Patricia.Davies@wlv.ac.uk**

If you would like specific information on the content of the MOOC then please contact the relevant instructor using the details below.

Java Programming

Dr. Patricia Davies	patricia.davies@wlv.ac.uk	Tel: 01902 328584

Maths for Programming

Dr. Liam Naughton	l.naughton@wlv.ac.uk	Tel: 01902 321452

Programming the Finch

Dr. Fernando Loizides	fernando.loizides@wlv.ac.uk	Tel: 01902 321443

We are happy to provide any assistance necessary, before, during and after completion of the MOOC.

Appendix

Detailed Course Syllabus and relationship to secondary syllabus.

Topic	Content	Skills	Link to GCSE & A-Level Syllabus
WEEK 0: **Introduction**	-MOOC overview -Getting set up -The java programming environment -Working with BlueJ		
WEEK 1: **Fundamental Java Concepts**	-Types and identifiers -Operators -Input/output -Comments -Documentation -Errors & exceptions	Design and implement computer computer-based solutions to problems in a variety of application areas	**Algorithms** • the use of data types: integer, real (double, float), Boolean, character & string. • abstraction. • decomposition. • algorithmic thinking. • how to produce algorithms using pseudocode and using flow diagrams. **Programming techniques** • the use of variables. constants, operators, inputs, outputs and assignments. • Casting. **Producing robust programs** • how to identify syntax and logic errors. • maintainability: comments and indentation. • the purpose of testing. **Computational Logic** • why data is represented in computer systems in binary form. • simple logic diagrams using the operations AND, OR and NOT. **[2.2.1 Data types] [2.3.1 Programming basics]** **[2.4.1 Programming] [2.4.2 Program testing]**

WEEK 2: **Control Structure**	-Decision-making (if-else, case) -Iteration (for and while loops) -Simple recursive methods	Develop an understanding of how computers make decisions, and how to write efficient programs that do repetitive tasks. Writing recursive methods. Understanding tail recursion.	**Algorithms** • how to interpret, correct or complete algorithms. **Programming techniques** • the use of the three basic programming constructs used to control the flow of a program: sequence, selection, iteration (count and condition controlled loops). **Producing robust programs** • defensive design considerations: input sanitisation/validation, planning for contingencies, anticipating misuse, authentication. • selecting and using suitable test data. **Computational Logic** • truth tables. • combining Boolean operators using AND, OR and NOT to two levels. • applying logical operators in appropriate truth tables to solve problems. **[2.3.2 Transferable skills] [2.3.3 Selection] [2.3.4 Iteration]**
WEEK 3: **Programming Projects-1**	-Introductory Labs	Writing simple Java Programs	
WEEK 4: **Arrays and Array Lists**	-One and Two - Dimensional Arrays -Array Lists	Initializing an array. Using arrays as parameters. Using array variables in a Class. Using an array of Class objects. Two-dimensional array algorithms. The ArrayList Class. The methods of	**Programming techniques** • how to define and use arrays as appropriate when solving problems. **[2.2.2 Arrays]** • use the technical terms associated with arrays including upper and lower bound. • select a suitable data structure (1D or 2D array) to use for a given task.

		ArrayList. Using ArrayLists.	• use pseudocode for 1D and 2D arrays (pseudocode will use square brackets to contain the array). • subscript, for example a 1D array as A[1:n] and a 2D array as C[1:m, 1:n]). • write program code using 1D and 2D arrays.
WEEK 5: **Sorting and Searching**	-Selection and Insertion Sorts -Sorting Algorithms that use Recursion -Recursive Sorts -Mergesort -Sequential/Linear Search -Binary Search	Implementing selection and insertion sorts. Analyzing recursive methods. Recursion in two-dimensional grids Implementing the mersgesort algorithm. Implementing sequential and binary search. Analyzing sort routines; best and worst cases. Analyzing search routines.	**Algorithms** • standard searching algorithms: binary search, linear search. • standard sorting algorithms: bubble sort, merge sort, insertion sort. **[2.2.2 Arrays]** • write algorithms/program code to process array data including: sorting using a bubble sort, searching using a linear search.
WEEK 6: **Object-Oriented Programming**	-Objects -Classes -Modifiers & Visibility: Public, Private, Static -Methods -Scope -References	Objects and Classes. Use public, private and static modifiers. Analyze methods and their use: header, types of methods, method overloading, data scope, the this keyword. Using method parameters. Understand the differences between passing by references and passing by object. Understand the use of the null reference.	**[2.3.6 Structured programming]** **[3.1.1 User-defined data types]** • show understanding of why user-defined types are necessary. • define and use non-composite types: enumerated, pointer. • define and use composite data types: set, record and class/object. • choose and design an appropriate user-defined data type for a given problem.
WEEK 7: **Some Standard Classes**	-The Object Class -The String Class -The Math Class -The Random Class	Understanding the universal superclass and its methods. Constructing String objects. Using the concatenation	**[2.3.5 Built-in functions]** • use a subset of the built-in functions and library routines supported by the chosen programming language. This should include those used for:

		operator. Comparing String Objects. Using other String methods.	• string/character manipulation. • formatting of numbers random number generator.
WEEK 8: **Programming Projects-2**	Advanced Labs	Design and development of programming projects.	