

A G.A.T.E.WAYS JOURNEY PROGRAM

for gifted Year 5 and 6 children with a

love of technology and coding

'Does It Compute?'

-learn to code with cards, dice

and ...Harry Potter

G.A.T.E.WAYS is an independent organization offering challenging and enriching activities and experiences to develop and extend highly able children. This *JOURNEY* for both girls and boys will run over four sessions.

Code is everywhere - it helps you look up interesting facts for your homework when you log in to Google; it enables you to watch YouTube clips and play games; it even helps your teacher or parents book your next G.A.T.E.WAYS adventure on the G.A.T.E.WAYS online portal. In this Journey you will start off with a number of hands on activities teaching you how to think like a computer and formulate algorithms using clear and unambiguous language. In the second half of the program we will learn to write our own code using Python, which is a text-based general-purpose programming language as opposed to a drag and drop coding program.

Requirements: Bring writing materials, a notebook, a blank USB to save your work in Session 3 and 4 a snack (no nuts please) and a small labelled photograph of yourself. Also bring a stamped, self-addressed DL envelope for your report. If you are attending this program in Term 1 or 4, you must bring a hat for the break

Session 1: Let's Think like a Computer!

A program essentially provides computers with a set of instructions to be obeyed. Computers follow instructions *exactly*, so that a small error in a code can cause a lot of problems. Imagine the consequences of an error in the coding for the latest driverless car. Intelligent machines require intelligent code. The first step to becoming a great coder involves *thinking* a bit like a computer. The idea is that we use language clearly and that we formulate our instructions in a logical sequence. In this session we will develop this skill via a range of activities: pairs of students will construct Lego with one person instructing and the other building. In groups we will work on a task management card grouping activity. Will your team be able to come up with the best method of utilising each team member just like a computer uses each processor? Finally, we will do some animated worded problems which have inherent imperfections and logical flaws, much like bugs in computer programs. The name of the game will be to find the bugs and optimise the solutions. By the end of this session you should be far more skilled at computational thinking.

Session 2: Creating our Own Algorithms

People at the cutting edge of mathematics and computer science go to work every day and grapple with the challenging problem of developing ever more efficient algorithms. (an algorithm is a set of steps that a human or a computer can follow to solve a problem.) Probably the most common problem of all is sorting, an operation that happens millions of times a second around the world in every email inbox, facebook profile and google search. So, there is a lot at stake in inventing new algorithms to do these operations ever more quickly. In this session we will look at several of these like bubble sort, quick sort, selection sort and insertion sort and play a game where we select ideal sorting algorithms for different data sets. We will look at two more classes of algorithm, the first dating back to ancient Greece which is a method to most quickly discover prime numbers and the second dating back to medieval Italy, a method to analyse games of chance and determine which, if any, are worth playing.

Session 3: Training Computers to solve problems!

Today we will begin by learning some basic concepts of Python programming, including *variables*, *loops* and *if statements*. We will put these together into programs that simulate conversations and do some basic mathematics. Next, we'll apply these concepts to writing algorithms that draw geometric shapes. This process will escalate from basic polygons to patterns like honeycomb, cubes and spirals. The session will conclude with us learning how to write functions in code that execute visual programs; the interesting part comes when these functions call on themselves. This effect results in self similar shapes which are called fractals. These occur everywhere in nature: trees, ferns, Romanesco broccoli and even river patterns.

Session 4: Harry Potter Meets Python Algorithms

In this session we are going to do something weird and wonderful!! We will import entire Harry Potter book into our program and then we are going to play around with text analyzing algorithms to find out new facts about Harry Potter. We could find which characters are mentioned the greatest number of times in the book, who interacts with whom the most and other fun trivia - what are like the longest words, longest sentences and the most common letters used. Before we are finished, we will also look at a field called *sentiment analysis* which can tell us both the mood (positive vs negative) of the text and its objectivity. The latter is a measure of how colourful versus how technical the text is. This can lead to some interesting revelations: we can discover the happiest chapters and characters and we can even try to determine heroes and villains by running an algorithm!

About the Presenter:

Sanjin Dedić is a robotics engineer with a background in product development and a teacher. His main passion in life is presenting programming and robotics with minimal academic jargon and abstract maths, in a way that can be embraced and understood by everyone, especially primary school students. In doing so he hopes that many younger more creative minds can join in the creation of the hi-tech world that is growing up around us.

