HyperionDev × UNISA ENTERPRISE

Fundamentals of Computer Science

MINI BOOTCAMP OVERVIEW

Overview

Welcome to the Fundamentals of Computer Science mini bootcamp, in partnership with UNISA Enterprise.

If you want to learn fundamental logical concepts that will be critical in all levels of programming and seek to acquire an understanding of computer science and software development, this bootcamp is for you.

To stay relevant and in-demand in the world of code requires that you're working with the most up-to-date tools. In this course, you'll be introduced to Python, the number one programming language as of 2020. You can expect to learn about control structures, looping, recursion, error handling and defensive programming as well as Object Oriented Programming to give you a holistic skillset highly sought after in the tech industry.

In this bootcamp, you'll be able to fast-track your computer science education in as little as 12 weeks. You'll develop a deep set of practical software engineering skills and will have the aid of expert code review. No previous coding skills are required.

The proccess

STEP 1

Log onto your personalised dashboard

STEP 2

Complete coding exercises online

STEP 3

Your code reviewer reviews your work within 48 hours

STEP 4

Perfect your coding over 12 weeks

STEP 5

Begin your new career in tech

Outcomes of this bootcamp

- Learn fundamental logical concepts that will be critical in all levels of programming
- Begin with the basic concepts of programming and logical thinking
- Acquire a foundational understanding of computer science

Our 1-on-1 code review centric approach works

Code review enables you to learn to code the right way through mastery of deeper aspects of software development that are a prerequisite for a career in coding. We help you master the deeper facets of industry-level development and set the foundation for a fulfilling career in coding. **Here's why learning through code review is smarter:**

DON'T MAKE THE SAME MISTAKES AS COMPUTERS

- Automated code checking is like spell check for computer programs. But you can't write a world-class essay with just good spelling — you need the right tone, facts, grammar, and style. Only human review of your code can help you learn aspects of coding that are analogous to tone and style that will make you truly fluent as a developer — automated graders just can't help you learn this!

GET UNSTUCK WITH ON-DEMAND TECHNICAL HELP

- Your code reviewer will ensure you move at a steady pace by helping you debug your programs within 48 hours. They will help you debug your programs and move forward so you never drop out.

BE EXPOSED TO THE INDUSTRY STANDARDS FROM DAY ONE

- Developers in the real world have their work assessed by a senior developer through the technique of code review. We're the only bootcamp in the world that exposes our students to this technique from day one so you get an advantage in the job market.

We layer a proven 1-on-1 personalised code review approach

INDUSTRY EXPERTS TAILORED TO YOUR GOALS

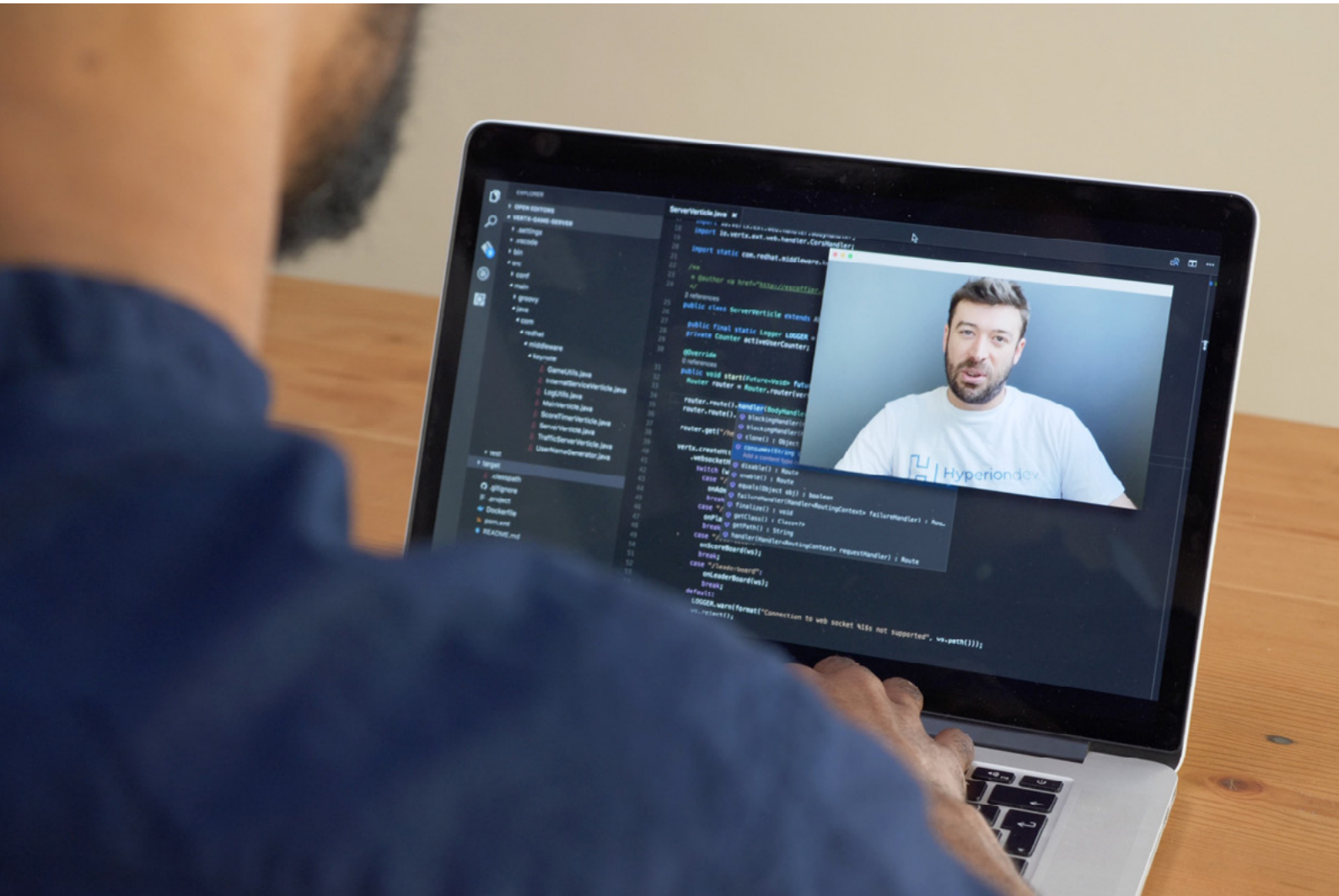
- You'll be paired with an experienced code reviewer who will guide you through 1:1 calls, career coaching, and live chat and email support.

JOIN A COMMUNITY OF CAREER-CHANGERS

- Learn as part of a cohort of students all working towards ultimate career fulfilment. Join online group tutorials, community chats and meetups, and peer coaching.

FREE OF FEAR OF FAILURE

- 1-on-1 guidance builds trust with your code reviewer and lets you progress at your own pace. Establish a safe space to discuss any roadblocks without fear of failure.



Why choose the Fundamentals of Computer Science mini bootcamp?

The comprehensive support structure and expert code review offered to you in this course is proven to get you coding better, faster, and entering the workplace skilled, confident, and prepared to hit the ground running. You'll begin with the basic concepts of programming and logical thinking, and then develop your career skill set with practical techniques in Python that cover the full spectrum of programming fundamentals. This opens many doors for you which we detail in the next section.

Career paths

THE SOFTWARE ENGINEER

- Write the code that makes the applications and software work for any major business across all different kinds of industries.

THE DEVOPS ENGINEER

- Combine IT, software engineering, and business management to decide what technology a business needs and how to implement it.

THE SOLUTIONS ARCHITECT

- Use a wide range of software, data, and engineering tools to build the systems that businesses use to make their work easier.

THE APPLICATIONS DEVELOPER

- Become the engineer who develops and delivers exactly what the client, business, or user wants, no matter its niche, needs, or design.

THE INFORMATION SYSTEMS MANAGER

- Use a wide set of software, and management principles to keep networks, machines, and technologies up and running in any business environment.

THE SYSTEMS ANALYST

- Combine technical software engineering skills, hardware management, and business logistics to create efficient and reliable systems.

Fundamentals of Computer Science

Duration: 3 months

Tasks: 29

Capstone projects: 4

The bootcamp is structured to allow you to start coding as soon as possible. Tasks are designed to:

- Teach you the theory needed to develop your skills.
- Give you the platform to practise implementing your new knowledge by completing one or more practical activities.

Remember, you're never alone. You can contact one of our expert code reviewers for 1:1 support whenever you need help with a task. The code you submit for each task is reviewed by a code reviewer who is an industry expert, to help improve efficiency and quality of code.

Breakdown of Syllabus

1	**Thinking like a programmer - Pseudo code I**	Learn how pseudo-code can help you clarify your thoughts and properly plan your programs before writing any code.
2	**Thinking like a programmer - Pseudo code II**	Delve further into algorithm design and representation.
3	**Your first computer program**	Get acquainted with Python, the powerful, easy to learn and extremely popular, high-level programming language.
4	**Variables - Storing data in programs**	Learn how to store and interact with the data in our programs using variables.
5	**The string data type**	Learn how to store and manipulate text using the String data type.
6	**Numerical data types**	Explore the different types of numbers used in the Python programming language.
7	**If Statements and the Boolean data type**	Learn how to use the if statement and the boolean data type to make decisions in your program.
8	**Beginner Control Structures - else Statements**	Beginner Control Structures - elif Statements
9	**Beginner Control Structures - elif Statements**	Learn how to check for multiple conditions using elif statements.

10	**Logical Programming - Operators**	Learn how to tell the compiler how to perform specific mathematical, relational or logical operations using operators.
11	**Capstone Project I - Variables and Control Structures**	Put your knowledge of variables and control structures to the test by creating an investment calculator.
12	**Beginner Control Structures - While Loop**	Learn how to execute a block of code repeatedly until a given condition returns false using while loops.
13	**Beginner Control Structures - For Loop**	Learn how to use the for loop to repeat a section of code a specified number of times.
14	**Towards Defensive Programming - Error Handling**	Discover the different types of errors that might occur in your programs and how to handle them.
15	**String Handling**	Learn how to manipulate text using Python's built-in functions.
16	**Beginner Data Structures - The List**	Discover the most frequently used and versatile collection data type used in Python - the list.
17	**Working with external data sources - Input**	Create smarter programs by learning how to read data from text files.
18	**Working with external data sources - Output**	Learn how to write data to text files.
19	**Capstone Project II - Files**	Put everything you've learnt about files to the test in this comprehensive task.

20	**Beginner Data Structures - Lists and Dictionaries**	Learn how to manipulate lists and become acquainted with dictionaries.
21	**Beginner programming with Functions - using built-in functions**	Learn how to use Python's built-in functions to provide better modularity for your programs and encourage code reuse.
22	**Beginner Programming with Functions - Defining Your Own Functions**	Create your own Python functions to carry out specific tasks.
23	**Capstone Project III - Lists, Functions and String Handling**	Use all the knowledge you have gained throughout this course to create a useful program.
24	**Introduction to Python - Data Structures - 2D Lists**	Discover the most frequently used and versatile collection data type used in Python - the list.
25	**Applied recursion**	Explore the concepts of recursive programming and how to “think recursively”.
26	**Towards defensive programming II**	Introduction to the principles of Object Oriented Programming
27	**Introduction to OOP I - Classes**	Introduction to the principles of Object Oriented Programming
28	**Introduction to OOP II - Inheritance**	Learn how you can improve the modularity and reuse of code using inheritance and the critical role it plays in Python's object system.
29	**Capstone Project - OOP**	Apply the fundamentals of object-orientation to solve a simple problem.