Curriculum Vitae – Alex Pearce

www.alexpearce.net
arpearce93@gmail.com
07763439025
D.O.B 15/07/1993

6 Lansdown Grove Lodge,
Bath.
BA1 5EJ.

Personal Statement

I am hard-working, reliable and eager to learn. I have many years of experience working in team orientated positions where communication is vital. I'm specifically looking to further my skill set within software development.

I have previously built applications with Java, Kotlin, PHP, Javascript, HTML, CSS and MySQL. Additionally, I have a good understanding of general object orientated principles, S.O.L.I.D and design patterns.

Education/Qualifications

BSc Computing
Birkbeck, University of London.

In July 2018, I completed a BSc Computing degree at the University of Birkbeck in London.
I achieved a 1st class honours degree.

10 A-C GCSEs.
Oakgrove Secondary School, Milton Keynes.

Grade A in Physical Education.
Grade B in Mathematics, Biology, Physics, Chemistry and Geography.
Grade C in English Literature, English Language, Humanities and Business Studies.

Recent Work History

Paxport, Bristol. August 2018 - Present.
Junior Software Developer.

Paxport provides a range of software products used in the travel industry. My role was to develop and maintain 'Find and Book'. 'Find and Book' is the aggregation service that organises many different services from flight, accommodation and ancillary suppliers into one single source. These suppliers include big travel industry entities such as Easyjet, Ryan Air and British Airways.

My time here has involved.

- Java 8 development.
- An agile style development methodology. Sprint based work.
- The use of Java tools and frameworks such as Maven, Spring boot, Dagger 2, Feign, JAXB, Mockito, WireMock and JUnit.
- Regular development against third party web APIs of many types. Restful, JSON, XML, SOAP etc.
- Performing code reviews.
- Working with continuous deployment systems. Specifically working with Gitlab CI and Jenkins.
- Deploying software to the Google Cloud platform. Docker containers were built from images

and deployed to Google Cloud's Kubernetes engine.
- Migration from a monolithic software architecture to microservices.
- Daily use of Jira as a ticketing and issue system.
- Working directly with Postgres databases to help troubleshoot and resolve bugs.

PricewaterhouseCoopers, Bristol. July-September 2017.
Data assurance intern.

I worked as part of the Data Assurance team in the PwC Bristol office. I was exposed to various technologies including Qlik Sense, the R programming language, Microsoft SQL Server and Aura Audit Software. The main project I was tasked with involved developing a dashboard within Qlik Sense to display utilisation of the Data Assurance team's hours. This meant creating a visually effective display of statistics and key performance indicators.

I also had the chance to join a separate team to test SOx (Sarbanes-Oxley) controls for Perrigo, an international PLC that manufactures pharmaceuticals.

Personal Projects

Spring Boot Instagram Service

I created a Spring Boot based web service which serves Instagram user data in a convenient JSON format. This data includes exact follower, following and post count, image urls and more. Multiple users can be scraped with a single request.

I chose Spring Boot because it's a fantastic tool for getting Java web applications up and running very quickly with little configuration. I used Maven for the project management tool and Jackson for dealing with JSON.

One of the challenges included scraping the Instagram pages in an efficient manner. I used an executor service to scrape separate users on different threads. This meant that response times were dramatically improved as opposed to working on a single thread.

Conway's Game Of Life

I re-created Conway's Game of Life in Java. This is a cell automaton where each individual cell has a state of being alive or dead. This state is dependent on its 8 neighbours. The rules are:

Any live cell with fewer than two live neighbors dies, as if by under population.
Any live cell with two or three live neighbors lives on to the next generation.
Any live cell with more than three live neighbors dies, as if by overpopulation.
Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

These rules lead to some really interesting re-occurring patterns.
One of the most challenging parts of this project was to develop an efficient algorithm for testing the state of each cell. I created the program to have varying cell sizes but the largest amount of cells possible at once is 480,000. I managed to come up with a way to only test "active cells". These are cells that are alive along with their 8 neighbours. By default, all other cells are dead. I also managed to selectively render the active cells. This lead to good performance.

For a complete list of projects, along with source code please visit my website listed at the top of this C.V.