

pcworkshopslondon.co.uk



Java 12 Weeks Boot Camp

pcworkshopslondon.co.uk

PCW Courses Ltd

t/a PCWorkshops

8 Northumberland Avenue

WC2N 5BY

Telephone: 020 7164 6359

Email: training@pcworkshopslondon.co.uk

PCW Courses Ltd. Trading As PCWorkshop

Company Number: 09990788

A 3D graphic consisting of several overlapping, semi-transparent rectangular planes in shades of light blue and grey. The planes are arranged to create a sense of depth and perspective. On the right side, a grey rectangular panel is prominently displayed, featuring the year "2020" in a large, bold, black sans-serif font.

2020

Java 12-Weeks Boot Camp

How does it work?	3
Duration:	3
Class times, 12 weeks, 1 lesson per week plus Self-study (+- 15 hours per week):.....	3
Number of lessons:	3
Self-study:	3
Self-study materials:	3
Evaluations:.....	3
Additional optional topics:.....	3
Projects:	3
Public port-folio:	3
Details:	3
Why attend the boot camp?	4
More about PCWorkshops:.....	4
What do you gain from the course?	5
CV and Job-search:.....	5
Examination and Certification:.....	5
Program:.....	6
Week 1: Java Programming Basics.....	7
Week 2: Object Orientated programming. Java Beyond Basics.	8
Week 3: Java Data Structures and Collections.....	9
Week 4: Database principles and SQL.....	10
Week 5: I/O Fundamentals / JDBC.....	11
Week 6: Java Test-Driven Development with JUnit.....	12
Week 7: API, Dates, OS, Scraping.....	13
Week 8: Spring & Hibernate.	14
Week 9: Java Concurrency and Multithreading.....	15
Week 10: Android Apps with the Android Studio.....	16
Week 11: Android Apps with the Android Studio.....	17
Week 12: HTML & CSS. JavaScript. PHP Basics.	18

The Java Coding boot camp

Our Mission with this course to give students fast-tracked entry to a programming/coding job. It suits someone who would prefer the fast-tracked route rather than a University degree.

- **Java In-depth**
- **Flexible**
- **Low-cost**
- **Lessons plus self-study**
- **Practicle**

Our vision:

Coding is a discipline that involves learning on an on-going basis. 3 months is a good period of time to skill up to the level of a junior coder, shorter may not be realistic. In these 3 months, we assist students all the way to an in-depth knowledge of the object orientated Java Language and all its aspects.

We inspire students to stay up to speed by regular quiz games and in-class tests.

We cover Java to a full depth, rather than touch lightly on many different topics.

We issue our own certificate for which there will be a theoretical and practical assessment at the end of the course.

Students will also be able to enter for the Oracle Examinations which are very well recognized in the industry. We assist with examples, mock tests, mentoring and extra tuition to prepare students.

Emphasis:

We immerse students in intensive training session and real-life practical examples. We dedicate a lot of time to practical exercise and projects. We believe that practical experience is the best instructor and serves best to get students ready for the workplace.

How does it work?

Duration: 3 months

Class times, 12 weeks, 1 lesson per week plus Self-study (+- 15 hours per week):

- Online or classroom, 11am – 6pm:
 - Fridays
 - Or
 - Saturdays
- Plus Self-study +- 15 hours per week (optional).

Number of lessons: 12, 1 lesson per week.

Self-study: Every week, 15 hours self-study recommended. Structured recommended self-study plan will be provided.

Self-study materials:

- Videos,
- manual,
- exercises,
- code examples,
- and tests.

Evaluations:

- Tests provided are practical, available in self-study materials.
- Optional. Students could opt to do a project instead.

Additional optional topics: Available with video, manual, exercises, code examples, and tests

Projects: Weekly exercises compound into a project.

Public port-folio: Projects are uploaded to Github and could be used to showcase to recruiters.

Details:

- **Study level:** Start from scratch through to job ready.
- **Virtual attendance:** Gotomeeting.com – no download required.
- **Classroom:** Golden Cross House, 8 Duncannon Street , London, WC2N 5BY.
- **What to bring:** Your own laptop.
- **Downloads:** Java JDK and Eclipse.
- **FAQ's:** <http://pcworkshopslondon.co.uk/faq.html>
- **Included:** Notes, Exercises, Online materials, Exams and Evaluations.

Why attend the boot camp?

Java affords you great career opportunities set to grow. Average London Salary: £65000. Coding jobs in high demand.

International trainers with extensive **real world experience**

Which means our courses are built on applicable real world scenarios and industry best practices (not simply theory).

A training business with a **consultative edge**

Extensive contact is given to each student, during and after the course. We take the time to truly understand the client's learning style and goals. We customize exercises and mentoring sessions around the student.

Our courses involve hands on **case studies and exercises**

Case studies and practical projects are tailored specifically to simulate current situations/difficulties that the client may be facing. Maximizing real life applicability of content is key to us, so exercises can be designed to help students work through issues step by step.

Training numbers are **capped at 15 people per course**

Integrity of content and delivery is essential and can often be compromised with large numbers. We cap our in-house training courses to a maximum of 15, so that all attendees remain engaged throughout and can take full advantage of the highly interactive learning environment.

More about PCWorkshops:

- PCWorkshops started as a Technology training company in 2013 in London.
- We have trained more than 2000 students in this time.
- Companies are technology companies, e-commerce, insurance companies, and banks
- Read more about us: <http://pcworkshops london.co.uk/>
- Locations: <http://pcworkshops london.co.uk/index.html#locations>
- Client Comments: <http://pcworkshops london.co.uk/index.html#Clients>
- Partners: <http://pcworkshops london.co.uk/index.html#partners>
- Clients: <http://pcworkshops london.co.uk/index.html#Clients>
- Course List: | <http://pcworkshops london.co.uk/Courses-categories.html>
- TandC's : <http://pcworkshops london.co.uk/Termsandconditions.html>

What do you gain from the course?

Skills & knowledge: Java programming knowledge and understanding with good practical application

Certification: Internal PCWorkshops certificate

Portfolio: You will have an online portfolio of Java projects

Experience: Our comprehensive practical exercises make you job ready

CV: We partner with recruiters to draft your CV professionally

Job Search: We assist with your job search and introduce you to employers

Oracle Exams: Exam Prep

Extras:

The British Computer Society: We will introduce you to the BCS. The Institute collaborates with government, industry and relevant bodies to establish good working practices, codes of conduct, skills frameworks and common standards. It also offers a range of consultancy services to employers to help them adopt best practice.

The Institute fosters links between experts from industry, academia and business to promote new thinking, education and knowledge sharing.

Membership: <http://www.bcs.org/category/5779>

Special Interest Groups: <http://www.bcs.org/category/5815>

BCS Events Calendar: <http://www.bcs.org/category/9231>

CV and Job-search:

- CV written by a professional
- Workshop: Attending an Interview
- Mock Interviews questions

Examination and Certification:

- You will get a PCWorkshops assessment-basic certificate
- You can on your own enter for the following certificate, for which we will provide exam-prep:
- Read more: https://education.oracle.com/java-se-11-programmer-i/pexam_1Z0-815
- We assist with examples, mock tests, mentoring and extra tuition to prepare students and with your registration for the exam.

Program:

- **Week 1: Java Programming Basics.**
- **Week 2: Object Orientated programming. Java Beyond Basics.**
- **Week 3: Java Data Structures and Collections,**
- **Week 4: Database principles and SQL.**
- **Week 5: I/O Fundamentals / JDBC.**
- **Week 6: Java Test-Driven Development with JUnit**
- **Week 7: API, Dates, OS, Scraping.**
- **Week 8 Spring & Hibernate.**
- **Week 9: Java Concurrency and Multithreading.**
- **Week 10: Android Apps with the Android Studio**
- **Week 11: Android Apps with the Android Studio**
- **Week 12: HTML & CSS. JavaScript, jQuery.**

Week 1: Java Programming Basics.

- **Week 1: Java Programming Basics.**
- **Week 2: Object Orientated programming. Java Beyond Basics.**
- **Week 3: Java Data Structures and Collections,**
- **Week 4: Database principles and SQL.**
- **Week 5: I/O Fundamentals / JDBC.**
- **Week 6: Java Test-Driven Development with JUnit**
- **Week 7: API, Dates, OS, Scraping.**
- **Week 8 Spring & Hibernate.**
- **Week 9: Java Concurrency and Multithreading.**
- **Week 10: Android Apps with the Android Studio**
- **Week 11: Android Apps with the Android Studio**
- **Week 12: HTML & CSS. JavaScript, jQuery.**

Java Programming Basics.

- **Overview of Java:** The three Object-Oriented Programming (OOP) principles;
- **Data Types, Variables, and Arrays:** Primitive types; Characters; Boolean; Working with variables and its scope; Type conversion and casting; Introduction to String class; Working with arrays
- **Operators and Expressions:** Introduction of operators; Arithmetic operators; Relational operators; Assignment operator; Logical operators; Increment and decrement operators; More in operators
- **Decision Making:** If statement; If - else statement; If- else if - else statement; Nested if - else; Switch Statements
- **Using Loops:** The while, do-while and the for loop; Enhanced for loop; Jump statements : break, continue; The return statement; Nesting loops; Using the for-each loop
- **Array Basics**
- **Using Methods:** Introduction of methods; Creating a Method in class; Calling a Method; Returning value from a Method; Adding a Method that takes parameters;

Week 2: Object Orientated programming. Java Beyond Basics.

- **Week 1: Java Programming Basics.**
- **Week 2: Object Orientated programming. Java Beyond Basics.**
- **Week 3: Java Data Structures and Collections,**
- **Week 4: Database principles and SQL.**
- **Week 5: I/O Fundamentals / JDBC.**
- **Week 6: Java Test-Driven Development with JUnit**
- **Week 7: API, Dates, OS, Scraping.**
- **Week 8 Spring & Hibernate.**
- **Week 9: Java Concurrency and Multithreading.**
- **Week 10: Android Apps with the Android Studio**
- **Week 11: Android Apps with the Android Studio**
- **Week 12: HTML & CSS. JavaScript, jQuery.**

OOP

- **Introduction to Classes and Objects:** Creating a Class; Creating an Object; Using an Object; Adding Instance variables; Controlling accessibility; Naming conventions for class members. Class Constructors; Parameterized Constructors; Inheritance. Abstraction. Interfaces and implementing interfaces. Override. Polymorphism. The dot operator, this keyword, the static keywords, the super keyword.
- **Object Orientated programming.** Java Beyond Basics.4 More in Methods and Classes: Overloading methods; Overloading Constructors; Using objects as parameters; Returning objects; Recursion;
- **Access control :** private, public and protected; The final keyword;
- **Encapsulation and Sub classing:** Using encapsulation in Java class design, Modeling business problems using Java classes, Making classes immutable, Creating and use Java subclasses, Overloading methods
- **Overriding Methods, Polymorphism, and Static Classes:** Using access levels: private, protected, default, and public., Overriding methods, Using virtual method invocation, Using varargs to specify variable arguments, Using the instance of operator to compare object types, Using upward and downward casts, Modeling business problems by using the static keyword, Implementing the singleton design pattern
- **Abstract and Nested Classes:** Designing general-purpose base classes by using abstract classes, Constructing abstract Java classes and subclasses, Applying final keyword in Java, Distinguish between top-level and nested classes
- Nested and inner classes; A stack class
- **Interfaces and Lambda Expressions:** Defining a Java interface, Choosing between interface inheritance and class inheritance, Extending an interface, Defaulting methods, Anonymous inner classes, Defining a Lambda Expression

Week 3: Java Data Structures and Collections.

- **Week 1: Java Programming Basics.**
- **Week 2: Object Orientated programming. Java Beyond Basics.**
- **Week 3: Java Data Structures and Collections,**
- **Week 4: Database principles and SQL.**
- **Week 5: I/O Fundamentals / JDBC.**
- **Week 6: Java Test-Driven Development with JUnit**
- **Week 7: API, Dates, OS, Scraping.**
- **Week 8 Spring & Hibernate.**
- **Week 9: Java Concurrency and Multithreading.**
- **Week 10: Android Apps with the Android Studio**
- **Week 11: Android Apps with the Android Studio**
- **Week 12: HTML & CSS. JavaScript, jQuery.**

Java Data Structures:

- **More about arrays.**
- **Collections: Lists, Sets, Maps, Trees.**

Week 4: Database principles and SQL.

- Week 1: Java Programming Basics.
- Week 2: Object Orientated programming. Java Beyond Basics.
- Week 3: Java Data Structures and Collections,
- Week 4: Database principles and SQL.
- Week 5: I/O Fundamentals / JDBC.
- Week 6: Java Test-Driven Development with JUnit
- Week 7: API, Dates, OS, Scraping.
- Week 8 Spring & Hibernate.
- Week 9: Java Concurrency and Multithreading.
- Week 10: Android Apps with the Android Studio
- Week 11: Android Apps with the Android Studio
- Week 12: HTML & CSS. JavaScript, jQuery.

SQL 1 Day Intermediate Course Description

Relational Database Concepts:

What is a Relational Database? Relationships.

Tables, Rows and Columns, Indexes, Primary Keys and Foreign Keys, Data Types

SQL Language Essentials:

The SQL Select Statement

SQL Conditions and the Where Clause: >, <, >=, <=, =, ,, Not, Between, AND, OR, Like, Wildcards

The SQL Order By Clause

SQL Arithmetic Operations, Expression Queries

SQL Column Aliases

Limit/Top, Distinct

Working with Null Values

SQL Summarizing and Grouping Data:

Aggregate Functions (Sum, Avg, Count, Max, Min)

The Group By Clause

The Having Clause

Querying Multiple Tables:

Joining Tables, Inner Joins, Outer Joins, Self Joins, Full Joins, Cross Joins

Alternative Join Syntax

Additional SQL Features:

Combining Queries

The Union, Intersect and Minus Operators

Basic Subqueries

- QL Queries, DML, DDL commands, stored procedures.
- Introduction to Big Data, NoSQL databases and Hadoop.

Week 5: I/O Fundamentals / JDBC.

- **Week 1: Java Programming Basics.**
- **Week 2: Object Orientated programming. Java Beyond Basics.**
- **Week 3: Java Data Structures and Collections,**
- **Week 4: Database principles and SQL.**
- **Week 5: I/O Fundamentals / JDBC.**
- **Week 6: Java Test-Driven Development with JUnit**
- **Week 7: API, Dates, OS, Scraping.**
- **Week 8 Spring & Hibernate.**
- **Week 9: Java Concurrency and Multithreading.**
- **Week 10: Android Apps with the Android Studio**
- **Week 11: Android Apps with the Android Studio**
- **Week 12: HTML & CSS. JavaScript, jQuery.**

Exception Handling and Files

- **Exception Handling:** Fundamentals of exception handling; Exception types; Using try and catch; Multiple catch clauses; Nested try statements; Throw and throws; The finally blocks
- **Exceptions and Assertions :** Defining the purpose of Java exceptions, Using the try and throw statements, Using the catch, multi-catch, and finally clauses, Multiple catch clauses; Nested try statements; Autoclose resources with a try-with-resources statement, Recognizing common exception classes and categories, Creating custom exceptions ; Throw and throws; The finally blocks. Testing invariants by using assertions
- **I/O Fundamentals, Input and Output streams. :** Describing the basics of input and output in Java, Read and write data from the console, Using streams to read and write files, Writing and read objects using serialization
- **File I/O (NIO.2):** Using the Path interface to operate on file and directory paths, Using the Files class to check, delete, copy, or move a file or directory, Using Stream API with NIO2
- **JDBC :** Connecting to databases and embedding SQL Queries to interact with database while coding, Defining the layout of the JDBC API, Connecting to a database by using a JDBC driver, Submitting queries and get results from the database, Specifying JDBC driver information externally,

Week 6: Java Test-Driven Development with JUnit.

- **Week 1: Java Programming Basics.**
- **Week 2: Object Orientated programming. Java Beyond Basics.**
- **Week 3: Java Data Structures and Collections,**
- **Week 4: Database principles and SQL.**
- **Week 5: I/O Fundamentals / JDBC.**
- **Week 6: Java Test-Driven Development with JUnit**
- **Week 7: API, Dates, OS, Scraping.**
- **Week 8 Spring & Hibernate.**
- **Week 9: Java Concurrency and Multithreading.**
- **Week 10: Android Apps with the Android Studio**
- **Week 11: Android Apps with the Android Studio**
- **Week 12: HTML & CSS. JavaScript, jQuery.**

Java JUnit Software Testing Course summary

- Overview: Different types of testing
- Principles of unit testing
- JUnit Environment Setup
- JUnit Test Framework
- JUnit Basic Usage
- JUnit API
- Writing a Tests
- Using Assertion
- Execution Procedure
- Executing Tests
- Suite Test
- Ignore Test
- Time Test
- Exceptions Test
- Parameterized Test
- Plug with Ant
- Plug with Eclipse
- JUnit Extensions

Week 7: API, Dates, OS, Scraping.

- **Week 1: Java Programming Basics.**
- **Week 2: Object Orientated programming. Java Beyond Basics.**
- **Week 3: Java Data Structures and Collections,**
- **Week 4: Database principles and SQL.**
- **Week 5: I/O Fundamentals / JDBC.**
- **Week 6: Java Test-Driven Development with JUnit**
- **Week 7: API, Dates, OS, Scraping.**
- **Week 8 Spring & Hibernate.**
- **Week 9: Java Concurrency and Multithreading.**
- **Week 10: Android Apps with the Android Studio**
- **Week 11: Android Apps with the Android Studio**
- **Week 12: HTML & CSS. JavaScript, jQuery.**

Api's

- **Generics, wrapper classes,**
- **Java Date/Time API:** Creating and manage date-based events and time-based events, Combining date and time into a single object, Working with dates and times across time zones, Managing changes resulting from daylight savings
- **Os Library**
- **Scrappy Library**

Week 8: Spring & Hibernate.

- **Week 1: Java Programming Basics.**
- **Week 2: Object Orientated programming. Java Beyond Basics.**
- **Week 3: Java Data Structures and Collections,**
- **Week 4: Database principles and SQL.**
- **Week 5: I/O Fundamentals / JDBC.**
- **Week 6: Java Test-Driven Development with JUnit**
- **Week 7: API, Dates, OS, Scraping.**
- **Week 8 Spring & Hibernate.**
- **Week 9: Java Concurrency and Multithreading.**
- **Week 10: Android Apps with the Android Studio**
- **Week 11: Android Apps with the Android Studio**
- **Week 12: HTML & CSS. JavaScript, jQuery.**

Spring and Hibernate frameworks

Spring, Hibernate and Struts are not a language, all these are frameworks that was used in Java Language. It helps to build mobile applications with Java framework. Spring is used to develop application from desktop to Web. Hibernate is used to access data layer and Struts is used for Web frameworks.

Week 9: Java Concurrency and Multithreading.

- Week 1: Java Programming Basics.
- Week 2: Object Orientated programming. Java Beyond Basics.
- Week 3: Java Data Structures and Collections,
- Week 4: Database principles and SQL.
- Week 5: I/O Fundamentals / JDBC.
- Week 6: Java Test-Driven Development with JUnit
- Week 7: API, Dates, OS, Scraping.
- Week 8 Spring & Hibernate.
- Week 9: Java Concurrency and Multithreading.
- Week 10: Android Apps with the Android Studio
- Week 11: Android Apps with the Android Studio
- Week 12: HTML & CSS. JavaScript, jQuery.

Java Concurrency and Multithreading.

- **Concurrency:**

Describing operating system task scheduling,
Creating worker threads using Runnable and Callable,
Using an ExecutorService to concurrently execute tasks,
Identifying potential threading problems,
Using synchronized and concurrent atomic to manage atomicity,
Using monitor locks to control the order of thread execution,
Using the java.util.concurrent collections,

- **Recursive Task**

Parallel Streams:
Reviewing the key characteristics of streams,
Describing how to make a stream pipeline execute in parallel,
List the key assumptions needed to use a parallel pipeline,
Defining reduction,
Describing why reduction requires an associative function,
Calculating a value using reduce,
Describing the process for decomposing and then merging work,
Listing the key performance considerations for parallel streams

Week 10: Android Apps with the Android Studio.

- **Week 1: Java Programming Basics.**
- **Week 2: Object Orientated programming. Java Beyond Basics.**
- **Week 3: Java Data Structures and Collections,**
- **Week 4: Database principles and SQL.**
- **Week 5: I/O Fundamentals / JDBC.**
- **Week 6: Java Test-Driven Development with JUnit**
- **Week 7: API, Dates, OS, Scraping.**
- **Week 8 Spring & Hibernate.**
- **Week 9: Java Concurrency and Multithreading.**
- **Week 10: Android Apps with the Android Studio**
- **Week 11: Android Apps with the Android Studio**
- **Week 12: HTML & CSS. JavaScript, jQuery.**

Android Studio , Creating app and designing Front-ends

Session 1: Android Studio Basics

Android Basics, Android - Environment Setup, Android - Application Components, Resources, Activities, Services, Broadcast Receivers, Content Providers, Fragments, Intents/Filters

Session 2: Creating User Interface with Android Studio

UI Layouts & UI Controls & Event Handling, Event Handling, Event listeners & handlers, Event Listener Registration, Using an Anonymous Inner Class, Using Activity Implements Listener Interface

Intents, Intent Objects, Android Intent Standard Action, Types of Intent, and Passing Data with Intent Extras.

Session 3: Android Studio Widgets

Basics: Structure of Android Application, The Activity File, The Manifest File, The Layout File, The Strings File Building a working application, Installing and Running on Device

Widgets UI Controls & Attributes: Receiving user input and responding,

UI Layouts, Android Layout Types, Linear Layout, Relative Layout, Layout Attributes.

Templates, Custom Fonts and Formats, Text View, Edit Text, Button, Checkbox, Toggle Button, Radio Group & Radio Button, Loading Spinners, Progress Circle, Progress Bar, Spelling Checker, Auto Complete.

Week 11: Android Apps with the Android Studio.

- Week 1: Java Programming Basics.
- Week 2: Object Orientated programming. Java Beyond Basics.
- Week 3: Java Data Structures and Collections,
- Week 4: Database principles and SQL.
- Week 5: I/O Fundamentals / JDBC.
- Week 6: Java Test-Driven Development with JUnit
- Week 7: API, Dates, OS, Scraping.
- Week 8 Spring & Hibernate.
- Week 9: Java Concurrency and Multithreading.
- Week 10: Android Apps with the Android Studio
- Week 11: Android Apps with the Android Studio
- Week 12: HTML & CSS. JavaScript, jQuery.

Android Studio , Creating app and designing Front-ends

Session 4: Android Studio Database interaction

Database: PHP/MySQL, SQLite Database, Create a program to sell movie tickets

Session 5: Android Studio Advanced Concepts

Sending Email, Sending SMS, and Phone Calls, PS: Find location, show location, and track

Session 6: Additional Android Studio Examples:

Localization, Google Maps, Bluetooth, NFC Guide, Wi-Fi

Week 12: HTML & CSS. JavaScript. PHP Basics.

- **Week 1: Java Programming Basics.**
- **Week 2: Object Orientated programming. Java Beyond Basics.**
- **Week 3: Java Data Structures and Collections,**
- **Week 4: Database principles and SQL.**
- **Week 5: I/O Fundamentals / JDBC.**
- **Week 6: Java Test-Driven Development with JUnit**
- **Week 7: API, Dates, OS, Scraping.**
- **Week 8 Spring & Hibernate.**
- **Week 9: Java Concurrency and Multithreading.**
- **Week 10: Android Apps with the Android Studio**
- **Week 11: Android Apps with the Android Studio**
- **Week 12: HTML & CSS. JavaScript, jQuery.**

HTML,CSS, Javascript and PHP

HTML and CSS: Basic tags

JavaScript: Responding to form end-user input

Php: Connect to a database using PHP