1 May 2018

The University of Manchester

School Of Computer Science

B.Sc. (Hons) Computer Science

Third Year Project

An eLearning Resource For A Computer Science Topic

Author:
Alex-Radu Malan

Supervisor:
Dr. Bijan Parsia

Abstract

The following report is discussing the third-year project of the author that is concerned about the available Java resources and their problem of delivering more advanced topics to the customers that are using their platforms. The discussion is focusing on available Java resources, potential gaps and limitations they all have, but also on the possibility of filling them with solutions also called objectives of the project. The justification on the decisions I have made, the tools and the resources used to accomplish the goal of the project are all covered in this report alongside a reflection, the difficulties that appeared through the course of the time and also a conclusion that will outline the accomplishments followed by possible improvements.

Acknowledgements

I would like to thank my supervisor Dr Bijan Parsia for being supportive of this project idea that I came up with. From the start, he was a really nice and fun person who taught me about different theories in teaching that would help me with the process of creating material for the project and also to gain a better understanding of something that I have not done before.

Special thanks to my colleague Adrian for sharing his thoughts and ideas about the project content and also to Andreea and my parents for being so supportive.

Table of Contents

Chapter 1

Context

Java is one of the most important programming languages used nowadays for creating web applications and platforms. A research conducted by TIOBE (14) showed the fact that Java is holding the first place on the most used programming languages in the world for the past fifteen years which is why it has such a big importance overall. The language itself has a motto of "write once, run everywhere" which is referring to the compilation into bytecode process of the actual code that will make it possible to be run on every JVM (Java Virtual Machine) (13). Nowadays most of the devices from laptops to smartphones and IoT devices can run Java. The number of Java online resources that are available today is countless, but only some of them are able to deliver course materials and practice content that would cover the core concepts in an interactive and motivational way for the users.

1.1 Aim

The aim of this project is to teach Java in a simple and effective way to the users that have reached a competent level of this programming language. The main purpose is to combine theory content explained easily with practice content in such a way that a person will be motivated to learn Java that is delivered in an interactive way rather than a non-interesting type of website.

1.2 Motivation

When somebody wants to acknowledge more complex Java topics, most of the online resources available have gaps in their course structure. A part of them have theory content displayed exactly like it is in a book (6) without practice content at all; other platforms deliver practice content, but a user is not able to complete them without the theory content which lacks; then some of the available resources have both practice and theory content delivered in an interactive way, but the difficulty level is limited to basic notions of Java which does not reach the goal of the person mentioned above. In order to have interactive, theory and practice content all-in-one, a new platform should be created as an answer to the existing gap. I have proposed this idea in order to gain knowledge about teaching people how to program in Java at competent levels and also in order to solve the gap problem most of the resources available have with an easy-to-use Java web tutorial.

1.3 Objectives

In order to achieve the goal proposed, there are three core objectives that have to be completed.

Theory content

In a matter of theory content, the website should be able to provide enough information about a certain concept and have a structure that will make the user understand terms and techniques from the simplest to the hardest ones. A good course will explain these terms in a way that the user will not feel overwhelmed or bored.

Practice Content

The practice content refers to the number of problems/exercises a user receives from a website so that the theory will be together with practice. By combining theory and practice, the user will be able to understand and have a better idea of the usability of the terms and techniques in the real world. If the user will practice coding intensively then he/she will get better and use the techniques and terms better than just reading them. The main idea is that the user will see and try to solve different problems and situations that are not possible in theory since the number of problems and their solving is infinite.

Interactive Design

When we talk about interactive design we refer to the fact that the website has a user-friendly design, easy-to-use and made in such a way that even the unexperienced user will know how to navigate and choose a path or a course. The user might receive trophies based on how much work and how much content he had completed. Examples of trophies might be even if the user is going to the website daily or at a certain interval, or even if the user is doing a daily challenge he/she might receive appreciation.

1.4 Overview

Chapter 2: In the following chapter we will discuss the background research that was necessary in order to find and compare the existing resources which lead us to a potential gap. Although there are also limitations that appear in this project since only one person is working on it in a limited amount of time.

Chapter 3: The chapter aims to provide information about the design of the platform, how the requirements are sorted based on the priority, the way users navigate through the application and the used technologies that lead to a working project.

Chapter 4: Implementation is one of the core parts of the project, so Chapter 4 will be about the materials and how they were created for the user, the way they are delivered and their order. There are also parts that will provide a timeline for the website construction.

Chapter 5: In the fifth chapter there will be a reflection on how was the process of creating the project, what difficulties and problems appeared and how they were solved in order to have everything on the go. There will be also an evaluation from the users perspective with feedback.

Chapter 6: The last chapter will have a conclusion on what has been achieved in the project, how it may be improved with new features and functionality but also what results were accomplished with it.

Chapter 2

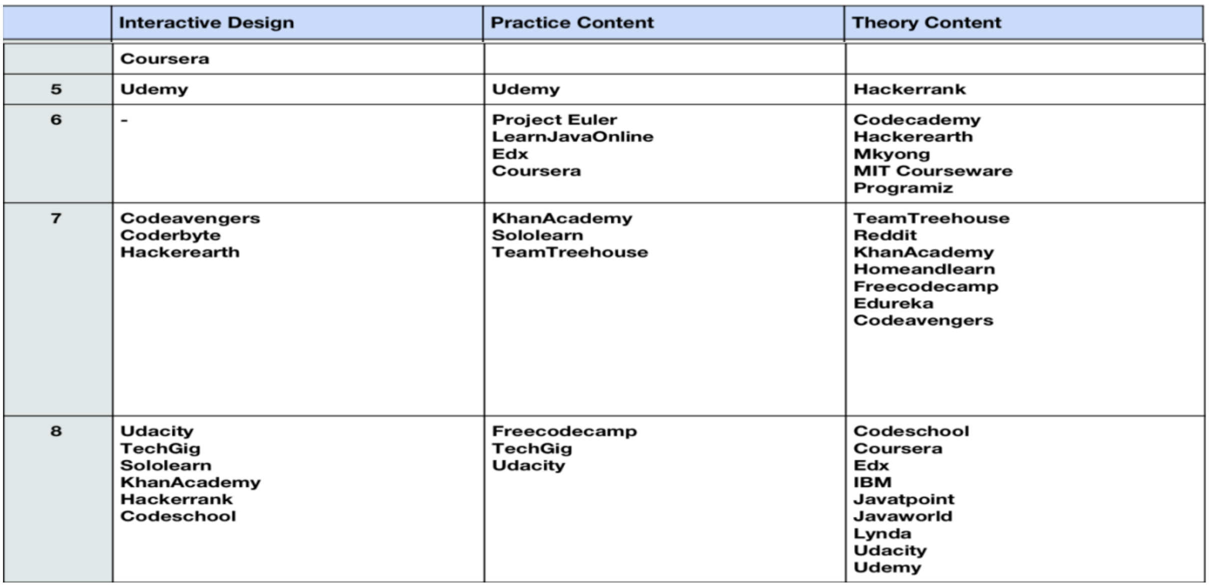
Background

Every idea has to be justified at a certain point. When the project proposal was made, there was no data, or any type of information related to the need of a new Java resource, only a partial idea based on testing multiple online resources for learning Java in the past. In order to have a strong proposal, I started to search and analyse the most popular and the highest rated Java resources available on the web. The idea was to compare multiple resources based on factors like: price, theory content, practice content, interactive content, difficulty and so on in order to determine which one of them or how many of them accomplish most of the user needs for quality theory, interactiveness that would create motivation and practice exercises that would test the user’s accumulated knowledge through the course (5).

2.1 Existing Resources

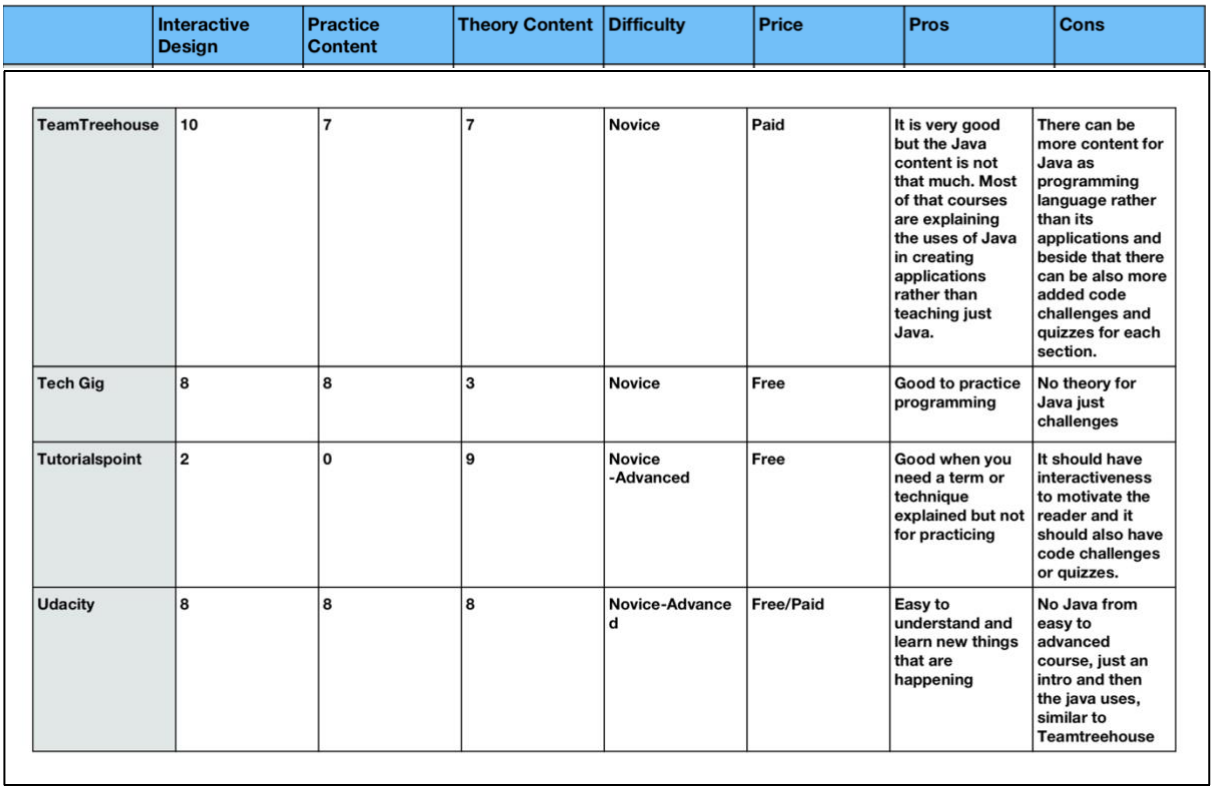
By searching on the internet, from the most basic types of searches to popular blogs and websites that would redirect the customers to those popular websites, a list of 49 resources was created. There were 49 websites of a different kind that would try to teach programming in multiple ways, each being better than the other. The factors described above were used and also a pro/cons type of comments was made for each of these resources. When the comments were finished, each website would get a grade in order for them to be ranked. The most common problem was the fact that most of these online resources were not even having Java materials, so a better filtering had to be done. After completing the process, a list of 40 most popular resources for Java learning was created, commented and ranked.

There are three core ranking factors: theory, interactive and practice content, which represents the most important parts of any massive open online course(MOOC) (1). Besides that, the difficulty used is ranging from beginner to novice and advanced. The pricing can also range from free to paid, and there are also free trial periods as well. In the end, the pros/cons are used to outline the strengths and the weaknesses of the website. The full research tables can be found in Appendix B.



	Interactive Design	Practice Content	Theory Content
	Coursera		
5	Udemy	Udemy	Hackerrank
6	-	Project Euler LearnJavaOnline Edx Coursera	Codecademy Hackerearth Mkyong MIT Courseware Programiz
7	Codeavengers Coderbyte Hackerearth	KhanAcademy Sololearn TeamTreehouse	TeamTreehouse Reddit KhanAcademy Homeandlearn Freecodecamp Edureka Codeavengers
8	Udacity TechGig Sololearn KhanAcademy Hackerrank Codeschool	Freecodecamp TechGig Udacity	Codeschool Coursera Edx IBM Javatpoint Javaworld Lynda Udacity Udemy

Figure 2.1 – Online Resources graded based on each factor described above



	Interactive Design	Practice Content	Theory Content	Difficulty	Price	Pros	Cons
TeamTreehouse	10	7	7	Novice	Paid	It is very good but the Java content is not that much. Most of that courses are explaining the uses of Java in creating applications rather than teaching just Java.	There can be more content for Java as programming language rather than its applications and beside that there can be also more added code challenges and quizzes for each section.
Tech Gig	8	8	3	Novice	Free	Good to practice programming	No theory for Java just challenges
Tutorialspoint	2	0	9	Novice -Advanced	Free	Good when you need a term or technique explained but not for practicing	It should have interactiveness to motivate the reader and it should also have code challenges or quizzes.
Udacity	8	8	8	Novice-Advance d	Free/Paid	Easy to understand and learn new things that are happening	No Java from easy to advanced course, just an intro and then the java uses, similar to Teamtreehouse

Figure 2.2 – Graded and commented resources with additional price and difficulty offered

2.2 Potential Gap

The results of the rankings showed that there are websites which provide a very good learning environment, content and practice delivered in an easy-to-use way. Those platforms in terms of numbers are very limited. After taking the best of the best, there was not a single resource that would be perfect in a matter of theory, practice and interactive content. The big gap outlined by the research showed the fact that most of these websites would provide only basic levels of Java programming theory followed by practice and interactiveness as well. The following websites are amongst the best ones found:

TeamTreehouse (15) is the leading platform with the highest score. In terms of theory, it teaches the basics of Java followed by tutorials on how to create different types of applications using this programming language. All the tutorials are delivered via videos featuring tutors who present the material in an interactive way using animations. The practice content is present in each video. The platform gives the user the possibility to skip the exercises which is a downside since the user should be able to complete it. Even though you get trophies of completion and interactive content, it is limited to the basics of Java.

Codecademy (2) has a different approach compared to TeamTreehouse. The theory is delivered on the left side of the screen while the practice is situated on the right side of it. Their idea is to have all in one so that the user would learn while doing which is an effective way of learning. Although they have badges and an easy-to-use platform, the content is also limited to the basics of Java.

Sololearn (12) is a platform designed to fit every device, from smartphones to desktops. The course covers almost all the Java theory followed by practice exercises after each theory chunk. The user receives a certificate of completion at the end of the course, but the problem with this resource is simply the fact that its practice content is a way to simple to complete. People with basic knowledge of Java can complete the code practice without even thinking of it.

Hackerank (4) is leading with the practice content since they have a platform made for those who want to practice and improve their coding skills. The theory on the other side lacks which would make Hackerank be unsuitable for the user that has little to no knowledge of Java. The website has interactive content, but it is not enough to provide only interactive and practice content without teaching a little bit of theory.

2.3 Goal

The project goal was to take this big gap found in all those websites and fill it in a simple manner. First of all, the theory content has to reach at least the competent level of Java, not just basic concepts, followed by enough practice content with levels ranging from ‘Easy’ to ‘Hard’. All the content and practice have to be delivered in an easy-to-use platform, where the user would be able to navigate intuitively without having to use the try-error design problem (10).

2.4 Limitations

In any of the platforms described above, there are teams of people that have way more resources and possibilities than a simple undergraduate student. The goal is not hard to formulate in theory, but there are limitations imposed not only by resources but also by time, that is why the project aims to achieve the best and core functionalities for the users rather than having additional functionalities which might be lower in a priority queue compared to those important factors.

Chapter 3

Design

The first steps into achieving the project goal are to start with an outline of how it will be created. This chapter is about the way requirements were established and also their priorities as well. Besides the specifications presented there are discussed the ways a certain user can navigate through the website and what technologies were used in order to make this platform available and existent.

3.1 Requirements

After researching the existing resources available to the public and comparing them based on the specified factors, some of the core requirements started to pop up quickly. The most important part was to prioritise them based on how effective would be in the learning process and how one can be more useful for the user even though it may not be that interesting or fun. The table below outlines the requirements ranked by their priority.

Priority	Requirement
High	I. Simple and easy-to-use platform that would eliminate any struggle for the user to interact with II. Course materials for the Competent Java user rather than just Java basic topics III. Practice content with different levels of difficulty in order for the user to apply the information from the lectures IV. Practice content in terms of code challenges V. Course materials to be presented in short-length videos(less than 10 minutes/video) in order to increase the interest of the user VI. Provide online compiler in order to make the practice exercises more interactive and fast VII. Provide further resources for the user(editors, frameworks, online resources, etc.)
Medium	I. Ability to sign up and log in on the platform II. Store the user progress and time spent on the course III. Trophies of completion based on the progress of the user IV. Java history presented in a timeline

	V. Practice projects for each completed chapter followed by a big overall project
Low	I. Create a compiler from scratch in order to store the user code II. Cover all the Java topics from basics to advanced III. Practice content in terms of quizzes and tests

Table 3.1 – Priorities and Requirements

The requirements found in the high priority section were all completed followed by two other requirements found in the medium priority section. Time was a constraint in this case, which made it impossible for all the requirements to be implemented. Although some of the functionalities would have been useful, there were other ones that can be classified as core functionalities. The ‘Difficulties’ sub-chapter is presenting unimplemented requirements and the reason for choosing them not to be in the final version of the project.

Most of the requirements listed in the high priority queue are meant to be completed in order to deliver a working platform with the most important and core features. The medium priority queue requirements can be implemented if there is enough time left after completing the first priority. Elements like the database, trophies for users, progress track are optional since the benefits a database can provide are low in our case. The code storage would be an option, having customisable profiles and access to own projects would all be benefits but those would also require a made from the scratch compiler that would deliver the code to the database. The compiler is time-consuming, so its priority has been set to low since the most important part of the project is to deliver course material followed by practice content.

The two elements from the medium priority queue are the timeline of Java history which is a great concept that would give the user a brief idea of the reason Java was created, its development and how it reached this popularity. The second requirement completed was the availability of a ‘Playground’ that would give the user the ability to create and try his/her own Java program ideas.

3.2 Navigation

One of the most important parts of every platform regardless of category is the ease of use. If the user will not be able to navigate intuitively through it, then there is for sure a design problem that has to be solved. For the project case, all the website pages are linked in such a manner so that people can navigate to any webpage regardless of the current page they are surfing on. In the figure below is an outline of the website structure.

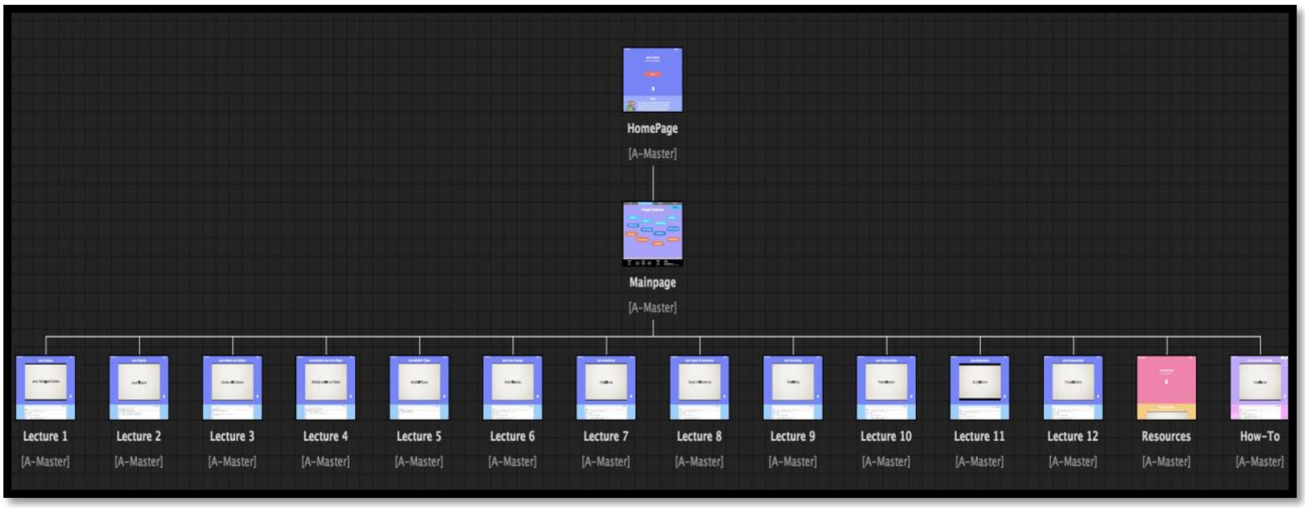


Figure 3.1 – Website diagram structure – The way everything is linked

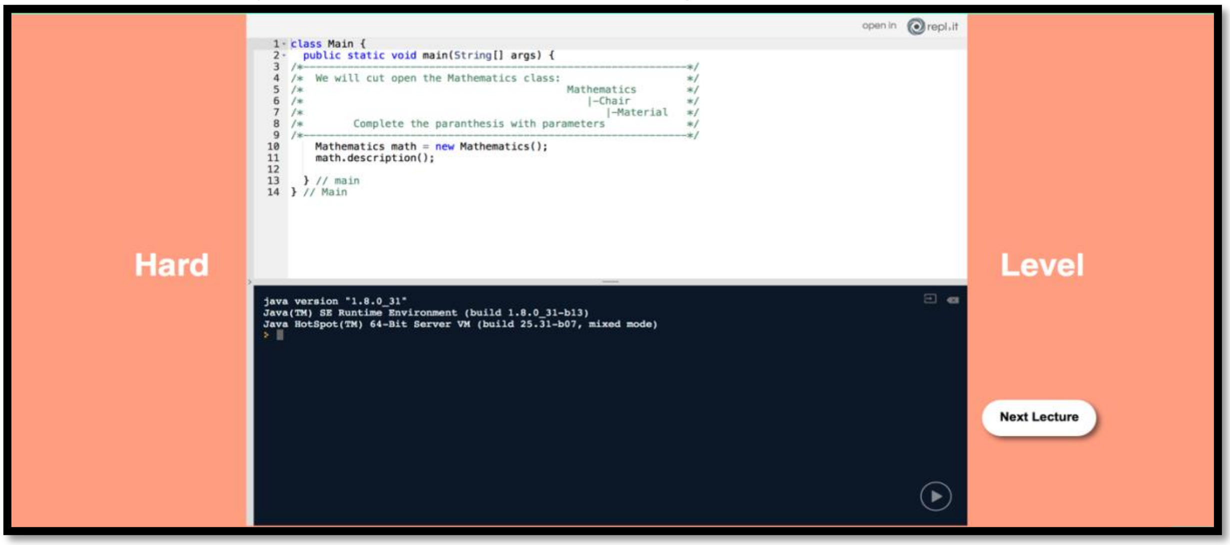
The first page that will appear would be the HomePage as seen on the diagram which would give the user an outline of the project and the features found inside. The second part of the HomePage would have the option to access the MainPage which is the control centre of all the other web pages. Each topic can be accessed from here and also the lectures are linked to each other in the order that the user should approach them. Besides the lectures, there are other two web pages that would provide further resources and a tutorial on how to use the website.

3.3 Technologies

Being a website dedicated to teaching and providing resources for a Computer Science topic, the technologies used are basic since the priority of the project was to have theory and practice material. The material and the way it should be presented took a big part of the time allocated for the project so the technical functionalities as seen in the above table were lowered in priority.

Practice content

The practice content made use of an embedded compiler – Repl.it(11) – which made it possible for the users to easily read and solve the code challenges.



```
class Main {
  public static void main(String[] args) {
/*------------------------------------------------------------*/
/*  We will cut open the Mathematics class:                    */
/*                                          Mathematics        */
/*                                            |-Chair          */
/*                                              |-Material     */
/*         Complete the paranthesis with parameters            */
/*------------------------------------------------------------*/
    Mathematics math = new Mathematics();
    math.description();

  } // main
} // Main
```

```
java version "1.8.0_31"
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
> 
```

Figure 3.2 – Practice content example from the Objects Topic

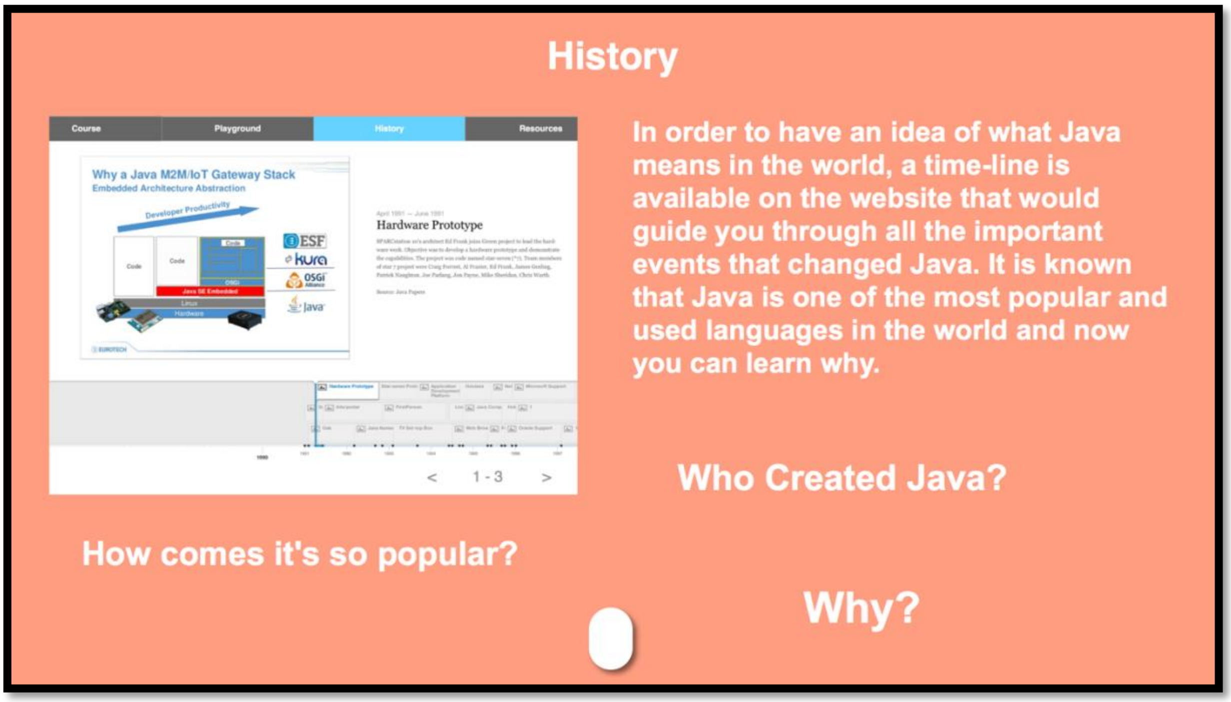
There are three code challenges for each Java topic with the difficulties ranging from easy to medium and hard so that the user will apply the knowledge gained from the lectures amongst the course.

Theory content

In terms of theory content, Microsoft PowerPoint(7) was the main tool used in order to create animations and provide structure to the content. To make it easier and more interactive, instead of using slides, the presentation was recorded using Apple QuickTime followed by video editing in iMovie which includes adding music so that nobody will have attention difficulties over a long period of time. The output of this process has been published on YouTube with ‘Unlisted’ Privacy Option and then embedded on the project website. The reason for using YouTube instead of directly publishing the videos on the website is simply because the YouTube player allows users to set the speed of the videos from 0.5X times to 2X times. The project targets the people with a competent level of Java, but there is no restriction for other users which may use the content for revision only.

Website

The website used HTML, CSS and JQUERY combined in order to give a better experience to the users and also to be as simple and interactive as possible. In order to gain more time in the process, Adobe Muse(8) has been used for the website design. Beside those technologies, Git Pages(3) is part of the project which is used as host provider, keeping it online for free. The other benefit of Git Pages was the fact that each time something is changed to the code, with a simple press of a button everything is sync and the website is updated in no time which made it convenient for multiple commits and changes over time.



History

In order to have an idea of what Java means in the world, a time-line is available on the website that would guide you through all the important events that changed Java. It is known that Java is one of the most popular and used languages in the world and now you can learn why.

Who Created Java?

How comes it's so popular?

Why?

Figure 3.3– The History section from the Home Page that would present the feature

Another benefit of Adobe Muse is the fact that the web pages can be optimised for each display type and dimension. From smartphones to monitors the project was made responsive in order to fit all the screens on which the course would be able to be taught onto.

Chapter 4

Implementation

This chapter aims to outline the steps that have been taken in order to achieve the course materials and their presentation to the user. The project is made out of three core parts: theory material, practice content and the website which is a host for the materials. There are also discussion of problems that appeared over the time and the decisions made in order to overcome them. Presentation of initial versus final versions of the website is also provided alongside explanations.

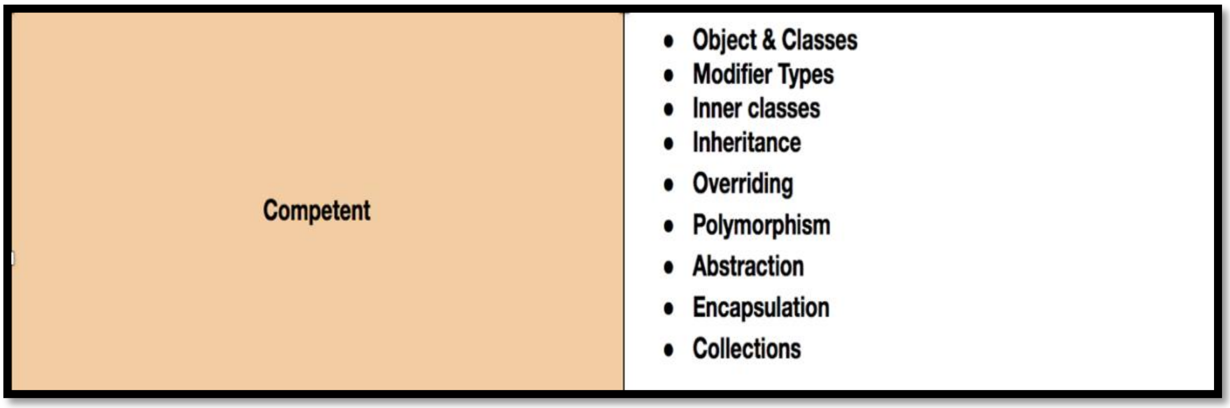
4.1 Course Material

One of the hardest parts of the project was to create and structure the course materials because I have never been teaching anything in the past. The first step was to decide what an intermediate user means, at what chapter is an intermediate user starting and how many chapters should I cover at least.

After conducting the research and also keeping an eye on what the other online Java resources are doing, I decided that the best chapter to start for an intermediate user is the "Classes and Objects". The basic user usually has a good understanding of variables, loops, statements and other elements that would give him/her the skills to create a simple program. Java classes and objects take the structure of the program to the next level, so that is the reason why this decision was made.

When it comes to talking about the number of chapters the course would present, I have started with two or three chapters since there were features such as progress bar and trophies of completion in the initial specifications. In the process, I have realised that it would be more important to allocate more time to the learning side rather than the technology one since the project is about an eLearning resource. The aim has changed from three chapters covered to at least six.

The second step would aim to create the structure of the course by deciding a path on which the user would go through. This would also mean the order of the course chapters that will be presented in the path.



Competent	• Object & Classes • Modifier Types • Inner classes • Inheritance • Overriding • Polymorphism • Abstraction • Encapsulation • Collections

Figure 4.1 – Chapters aimed to be covered in the project(9, 16)

After having everything set up, the target was to create a comparison with a real-life example which in this case would have been a school with multiple classrooms inside which would represent the multitude of classes that linked to each other form a program as a whole. All those details were put up in a script which can be compared to a story (Appendix A).

After finishing the script, I have chosen to present it using animations in order to keep everything simple. There were problems that appeared such as the lack of experience that I had, the fact that most of the animations tools required the user to pay a fee which was high, but after all the obstacles the best tool was Microsoft PowerPoint. The script was transposed into slides with simple animations that would explain the course in an interactive way.

The idea behind the course material is to be delivered in form of a video to the user of the website. Slides and their animations were recorded using Apple QuickTime and then edited in iMovie in order to have sound in it. The song used in the videos is copyright free and with a genre similar to jazz so that the user will not lose interest over the period of the course.

4.2 Practice Content

In the initial requirements, the practice content is related to quizzes and tests that the user should undertake in order to test his/her knowledge. After analysing the progress and the time a quiz is taking, I have realised that a better idea was to use code challenges that would suit the story of the content. Previously there was specified the fact that the course story is about a school with multiple classrooms which are related to Java classes. Users are required to create those classes by defining them and also giving them different behaviour based on each of their subjects.

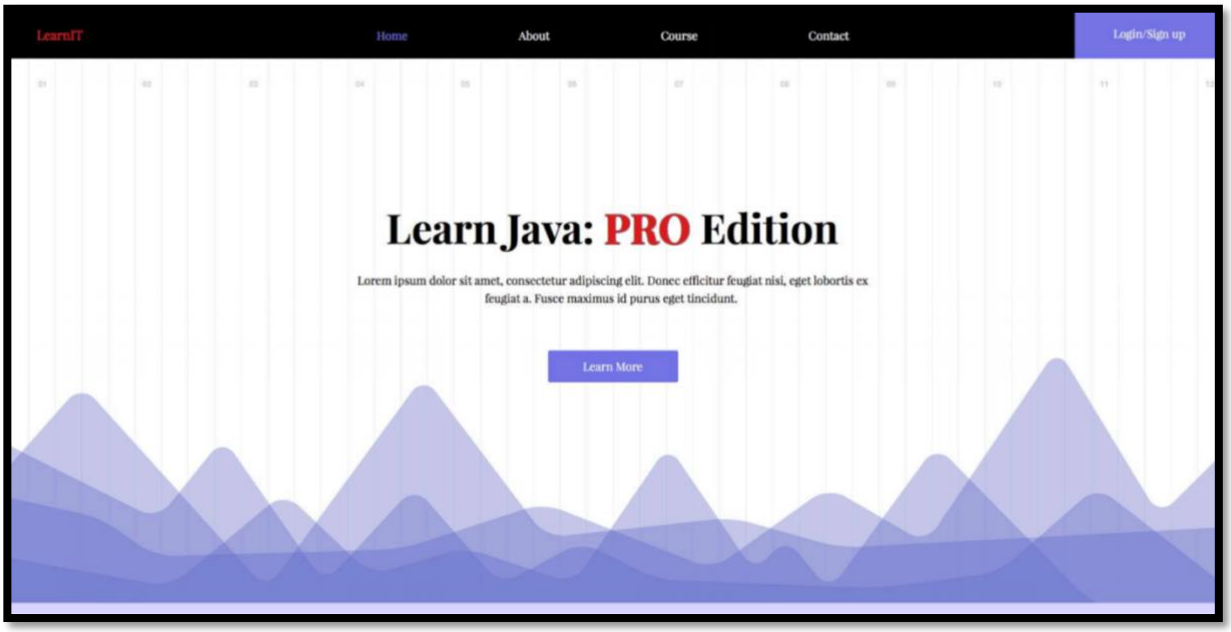
In order to provide a tool that would require the user only to write the code and then execute it by the press of a button, the website has to have a compiler. Another discussion was about the worthiness of implementing a compiler from scratch which would take more time and also the functionalities after all would not be justifiable. The benefits of having a compiler would be the fact that all the user code would be checked in the background by using test cases and then print out messages based on what error the user has made. Amongst other features would be the code storage for each person which is also a downside to the learning process. People would log into their accounts and when they would navigate to a course chapter that has been finished, the code would be there, which is something Codecademy is doing.

The idea behind the practice content is to try to solve the code in a loop rather than just doing the exercise and then never check it back again. The compiler's priority was set to low since the time had to be used for content creation and also providing an interactive platform to the user.

Instead of creating a compiler from scratch, the website has embedded a ready-to-use compiler provided by REPL.IT. For each topic there are three projects created on reply.it, so there are multiple embedded projects inside the website with the shallow code inside in order for the user to complete it. Above there was specified that three practice exercises are provided for each course topic since there are three difficulties: easy, medium, hard.

4.3 Website

For the platform, everything started with a mock-up that would try to give an outline of how the website should look, which changed in the implementation.

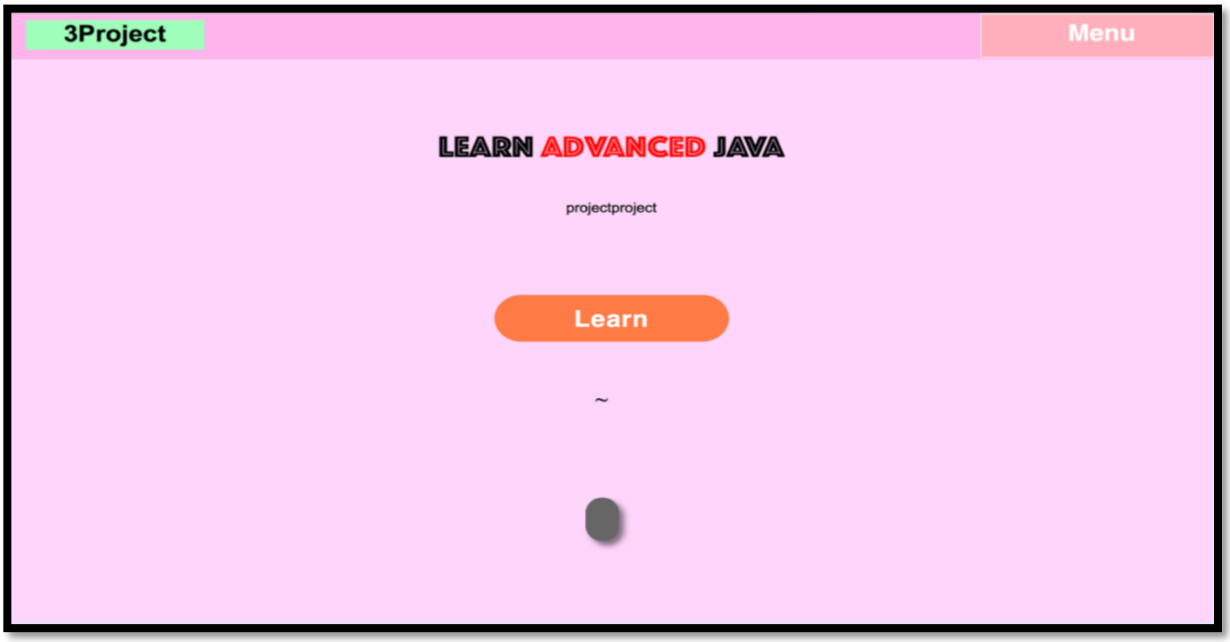


Learn Java: PRO Edition

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec efficitur feugiat nisi, eget lobortis ex feugiat a. Fusce maximus id purus eget tincidunt.

Figure 4.2 – Initial mock-up of the website

Over time the design has changed, and different tools were used to implement it, such as Adobe Muse, which was the main creation tool used for the platform. Besides the features that give the developer the power to create a website without refreshing the browser multiple times to check if the elements are in order. It also provided tools for embedding the course material and the practice content into the website so that it will run smooth and fast without the need of loading pages for each action taken by the user.



LEARN ADVANCED JAVA

projectproject

Figure 4.3 – First version of the website created in Adobe Muse

In Figure 1.8 the first version of the website was made. In order to make it as simple and easy as possible, so in Figure 1.9 the final version shows how the use of a single colour to mask all the other boxes can output a way easier and user-friendly website.

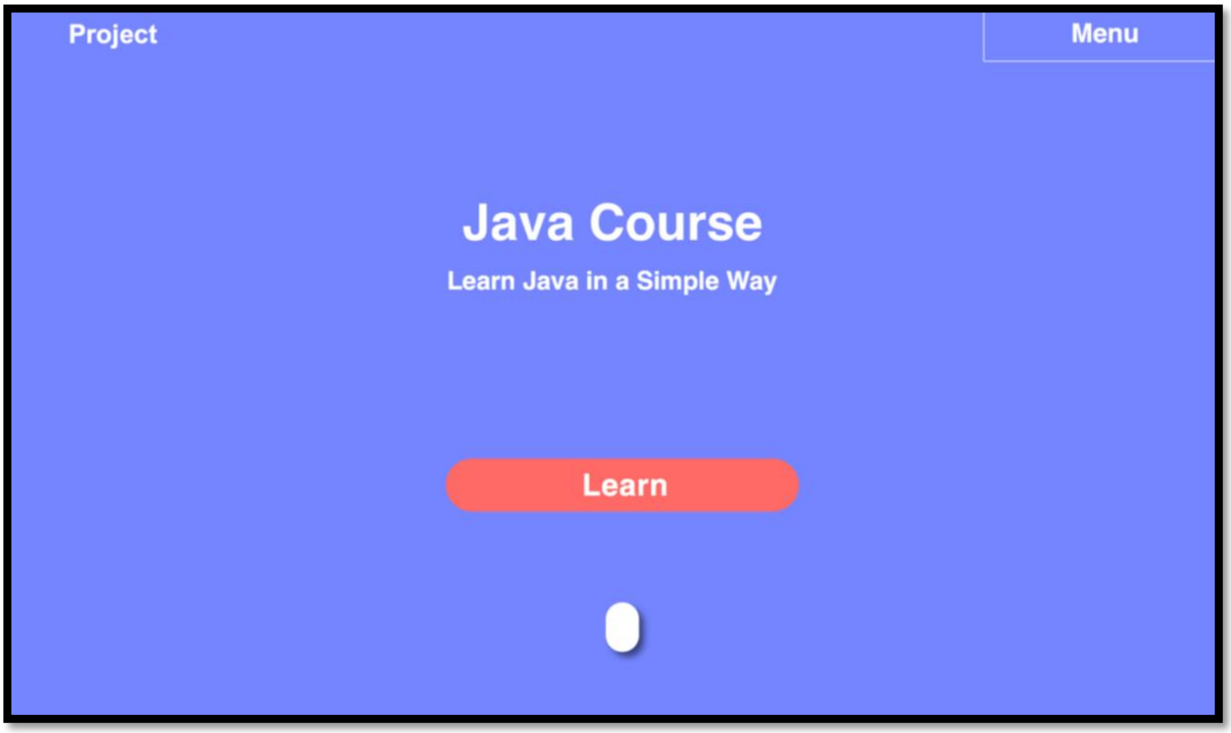


Figure 4.4 – Final version of the website – the home page is presented above

Chapter 5

Reflection

Looking back at the beginning of the project most of the features and plans were too optimistic. The first idea was to create a Java course that will cover all the topics in the programming language which is not possible given the timeframe available. Moreover, the course was planned to be hosted on a platform which will provide all the benefits a normal online Java resource would since the goal was to surpass the competition and get a 10/10 in the grading table for all the three factors (theory content, practice content, interactiveness). Over the time I have learned more about teaching and how much time is taking the course material creation process, which back then, in my opinion, was very simple until I started to work on it. The person conceiving these materials has to put himself/herself in the place of the learner and find the best way he/she can deliver the content. There are multiple ways to do this, but the most appealing to me was the comparison idea in which all the content is spinning around a story about students in classrooms and the management system of a school. I found it challenging to create the course and it took a while to accommodate with the Microsoft PowerPoint animations since this is the few times that I have used it for animations that are more complex than just some up and down transitions. The first lecture took about a week to be done in the application with all the animations and the content which made me nervous at the time since the goal was to have at least five to seven lectures at least. The time passed, and my skills were improving after each slide which helped me to achieve twelve lectures in which were presented ten Java chapters. This part of the project took the most time to complete, which was a big setback for the practice content due to the time constraints. I found myself writing forty classes of Java per day just to complete at least half of the course practice requirements since there are twelve lectures with three code challenges for each of them and those challenges have about four to five Java classes each.

This was an interesting project that got my interest from the start and I am really glad about the skills that I have gained while creating it. Even though it is not having the features and functionalities that those with the highest rank in my list have, but it achieved the goal of delivering the core material to the user.

5.1 Difficulties

In the table above, there were requirements with lower priority and also medium. The first part of the discussion of difficulties will be concerned about the medium priority queue.

First of all, the project was having an initial plan that was unrealistic because of the lack of experience in the teaching domain and the estimations that were not accurately measured from the start. The most important part of the project was to have an interactive platform with theory enough to cover an intermediate level of Java followed by practice content. This part described here is concerned more with the learning process rather than the features a user has. The medium priority requirements proposed to have a database with a user management system that would allow the users to sign up and then log into their accounts. This functionality would require a significant amount of time and also there would enough benefits that would justify the time spent on such a task. By having a database, besides the profile that can be customised by each user, there would be also a progress bar that would point the last chapter a user has completed and there would be trophies of completion for the topics finished and the timeframe in which those were done. When the high priority queue was completed, I tried to set up a database using Firebase service that would provide database services such as authorisation, user management, 2FA, cloud database, and others. It is straightforward to do the authentication process, but the other parts that would require the "tables" which are in JSON form would take more time to have links between HTML boxes and those. Another problem would have been the fact that Adobe Muse output code is messy which increase the difficulty of implementing such a feature. It is not impossible, but the time constraint made it impossible to be implemented. Another feature in terms of education would have been the big projects at the end of a certain group of topics which would encourage the user to create a Java program that would have real-world application even if it would be simple. The problem with implementing and creating such a feature is simply because of the compiler that is set to the lowest priority since it is complex and time-consuming to have a made from the scratch compiler.

The final part of the difficulties in linking the lowest priorities in the requirements table. The most important part of this section is the compiler made from scratch. This is a complex problem that would require way too much time in order to be justified. There are benefits such as customised messages for each user by congratulating them or sending them a simple 'Bummer' message when an error is found in the code followed by test cases, code storage and other functionalities. Those would have been great for the user but given the fact that real.it has a really great and easy-to-use compiler that can be embedded, made me decide that in terms of time management it would be a better choice compared to creating a compiler from scratch. Other requirements presented there are also the tests and quizzes that would be provided beside the code challenges. Time was again a problem in this case simply because at the start of the project I taught that a quiz is very easy to create and implement on the website which proved to be wrong. Quizzes take a long time just because of their change of revealing answers to questions or even providing questions that would test the knowledge of a person rather than just the memory. These are some reasons why the tests and quizzes were dropped for the final version of the project.

Chapter 6

Conclusions

The chapter discusses the overall progress, the achievements and further improvements that can be implemented into the project with everything being summarised as well in the following subsection.

6.1 Results

The goal of the project was to deliver a simple and easy-to-use Java online resource that would cover an intermediate level of Java topics in videos followed by practice content with ranging difficulty. This goal was achieved, with some features that are still unimplemented such as the database side, the compiler and the quizzes to provide further practice. There were more topics covered than in the initial plan, the practice content for each of them is completed as well and also the platform has achieved its goal more than expected through its design and simplicity.

6.2 Future Improvements

The project goal as mentioned above was achieved with some functionalities still being in a queue and waiting to be implemented, but the complexity and resources needed for them to be completed are not hard to be achieved. The only resource needed in order to create them is time. Besides those requirements, there are multiple possibilities for future improvements such as leaderboards, forums, a chat that will allow the platform to have a community of people helping each other in solving problems. Also, the practice content can be multiplied and diversified as well so that the users would have the ability to test their knowledge in different ways. The resources provided to the people can also be more diversified and the course can cover more than just the intermediate level of Java. All the topics can be covered with further lessons concerning frameworks and their application in the real world. Big projects that would help users understand how to use Java for client-server programs and other parts as well. The improvement area cannot be defined since there are countless ways the platform can be improved.

6.3 Summary

This report aimed to provide a detailed explanation of the process behind the development of an eLearning Resource for a Computer Science Topic which took the form of a Java tutorial. The platform was developed at The University of Manchester as my third-year project at the School of Computer Science. The report explained the decisions made during the development, the details concerning the existing online resources, the gap that was partially filled by the project which was not made possible by the other Java websites and also a reflection on the process of learning and creating the materials that play a core role in achieving the goal of the project. After all, it was a great experience from which I learned more about teaching and the management of different problems that may arise in the process of developing a product.

Reference

1. Anon, (2013). *Massive Open Online Courses, aka MOOCs, Transform Higher Education and Science*. [online] Available at: http://er.dut.ac.za/bitstream/handle/123456789/70/Waldrop_2013_MOOCs_transform_HE_and_Science_Scientific_American.pdf?sequence=1&isAllowed=y

2. Codecademy. (n.d.). *Codecademy - learn to code, interactively, for free*. [online] Available at: https://www.codecademy.com

3. GitHub Pages. (n.d.). *GitHub Pages*. [online] Available at: https://pages.github.com

4. HackerRank. (n.d.). *HackerRank*. [online] Available at: https://www.hackerrank.com

5. Kim, A. and Ko, A. (n.d.). *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education - SIGCSE '17*. [online] Available at: https://faculty.washington.edu/ajko/papers/Kim2017CodingTutorialEvaluation.pdf

6. Latham, J. (2015). *Java: Just in Time*. London: College Publications.---------

7. Microsoft.com. (n.d.). *PowerPoint 2016 - Microsoft Store UK*. [online] Available at: https://www.microsoft.com/en-gb/store/b/powerpoint-2016?invsrc=search&cl_vend=google&cl_ch=sem&cl_camp=973816545&cl_adg=52087117161&cl_crtv=230574081797&cl_kw=powerpoint&cl_pub=google.com&cl_place=&cl_dvt=c&cl_pos=1t1&cl_mt=e&cl_gtid=kwd-11651171&cl_pltr=&cl_dim0=Wug66AAABYrrOsra:20180501100112:s&OCID=AID695748_SEM_Wug66AAABYrrOsra:20180501100112:s&s_kwcid=AL!4249!3!230574081797!e!!g!!powerpoint&ef_id=Wug66AAABYrrOsra:20180501100112:s

8. Muse.adobe.com. (n.d.). *What's New | Adobe Muse CC*. [online] Available at: http://muse.adobe.com

9. Niemeyer, P. and Knudsen, J. (2002). *Learning Java*. Sebastopol, CA: O'Reilly.

10. Norman, D. (2002). *The design of everyday things*. New York: Basic Books.

11. repl.it. (n.d.). *Online REPL, Compiler & IDE*. [online] Available at: https://repl.it

12. SoloLearn. (n.d.). *SoloLearn: Learn to Code*. [online] Available at: https://www.sololearn.com

13. TheServerSide.com. (2016). *Why Java is the most popular programming language*. [online] Available at: https://www.theserverside.com/feature/Why-Java-is-the-most-popular-programming-language

14. Tiobe.com. (n.d.). *TIOBE Index | TIOBE - The Software Quality Company*. [online] Available at: https://www.tiobe.com/tiobe-index/

15. Treehouse. (n.d.). *Start Learning at Treehouse for Free*. [online] Available at: https://teamtreehouse.com [Accessed 1 May 2018].

16. www.tutorialspoint.com. (n.d.). *Java Tutorial*. [online] Available at: https://www.tutorialspoint.com/java/index.htm

Appendix

Appendix A

The contents of this appendix refer to the script made during the process of course materials development. The material might contain grammar mistakes or uncompleted sections due to the fact that it is a draft.

Objects and Classes

In order to create a program there are 2 ways:

1. Using one class
2. Using multiple classes that are linked to each other

Think of this as having a school
A school teach multiple subjects, not just one!

What is the best option?

1. Put all the people in a single classroom - huge one
2. Split the people in smaller groups then set a timetable and put them in more classrooms which would increase the engagement and interactiveness between teacher and students

Which one is more efficient?What are the pros/cons?
If we put all the students in one room then if there are questions it will take long to answer all of them and maybe the others would want to go home not stay another hour just for Q/A
If we split them then it is tricky to do the timetables for each group - in the idea that unlimited amount of students can attend at any time of the year - no starting/ending date

After you learn the basics of Java programming you should know the components that create a program. Let’s learn how can we use them to create one!

First, what is a class?

In our school example, let’s say that each classroom has a label on the door with the subject taught in that classroom, so students would move around after each hour and search for their next class/subject in their timetable(ex. class Mathematics).
In order to identify the classes and make use of them in programming, we need to label/name them.

What elements are in a class?

A class can contain different type of elements, from variables to methods which have more categories

How do we define a class in programming?

We can simply just write class and then give it a name.
The file in which we write the code has to have the same name as the class defined in it.

A quick example:

```
-----------------------------------------------------------------
Mathematics.java
-----------------------------------------------------------------
public class Mathematics {
        …...
}
-----------------------------------------------------------------
```

First, what is in a class?

Local variables - as you might already know the local variables are the ones inside a method, a block or in a constructor(we will talk about this in a minute)
For example:

```
-----------------------------------------------------------------
```

```
Mathematics.java
-------------------------------------------------------------------
public class Mathematics {
        public static void main(String args[]) {
                int myVariable = 8;
           ……
      }
}
-------------------------------------------------------------------
```

‘myVariable’ is a classic local variable example, it can have any type of data type from byte, short, int to char.

Instance variables

These types of variables are the opposite to the local variables since they are declared outside any method, block or constructor.

```
-------------------------------------------------------------------
Mathematics.java
-------------------------------------------------------------------
public class Mathematics {
        int seatsAvailable;
        int booksAvailable;
        String teacherName;
 }
-------------------------------------------------------------------
```

In the mathematics classroom we have to know how many seats are there, books so that students will have one on each table at least, the teacher’s name, etc.

What do we do with these instance variables?

We use them in methods called **instance methods**. These methods make use of the variables in order to return a certain result based on each method.

```
--------------------------------------------------------------
Mathematics.java
--------------------------------------------------------------
public class Mathematics {
        int seatsAvailable;
        int booksAvailable;
        String teacherName;

        public void getTeacherName()
        {
                System.out.println(“The mathematics teacher is called” +
teacherName);
        }
 }
--------------------------------------------------------------
```

In this instance method we want to retrieve the mathematics teacher’s name. What can be the problem? We have the variable declared, the method as well, but how do we set a name to that teacher?

Constructor Method

Before we declare the instance methods and after the instance variables declaration we have to create a constructor method. This is going to be the control center which will assign each variable with a value.

```
--------------------------------------------------------------
 Mathematics.java
--------------------------------------------------------------
public class Mathematics {
        int seatsAvailable;
        String teacherName;

        public Mathematics(int seats, String name) {
                seatsAvailable = seats;
                teacherName = name;
```

```
    }

    public void getSeatsAvailable()
    {
        System.out.println(“The number of seats available in the
    Mathematics classroom is ” + seatsAvailable);
    }

    public void getTeacherName()
    {
        System.out.println(“The mathematics teacher is called” +
    teacherName);
    }

 }
```

Basically the constructor method will have 3 parameters, each of them being assigned to the instance variables declared above the constructor method. After this assign action then the variables will be able to be used in the instance methods…. But how?

Java Objects

So far the concept of class is good for organising a program so when a problem appears the programmer would search for the class related to that problem instead of going through 100.000+ lines of code, but what is a class in essence?

Let’s stick to the school example. In the school there are tons of objects, from pencils to tables to students and everything that is existing in it(students count as objects).
Each object has properties and behaviour. For example, the students, each one of them can be seen as an object that has different properties/looks to behaviour.

A class is basically a template that would be used to create objects.
For the mathematics classroom we have so far the number of seats and the teacher’s name which represent some of the **classroom properties**.

The mathematics classroom is basically an object in the school. Here we can see a hierarchy of classes in the school case scenario:

class School

class Mathematics   class Biology   class Spanish ….

Pens, seats, tables, students, books, etc.

In programming terms, a School class is the root of every other class(room) simply because each classroom from Mathematics to Biology, Spanish etc are included into it. All these classrooms have seats available, books available teachers and their name, pens, so common properties. Each of these classes is an object in the class School.

So in the end we have a ‘class School’ with different topic objects that has to be created in there. How?

First of all let’s take a look back at the example with the complete Mathematics class. In order to get the result for the teacher name we need to supply the instance variable with a name.

```
---------------------------------------------------------------
School.java
---------------------------------------------------------------
public class School {
        public static void main(String args[]) {
        Mathematics mathClassroom = new Mathematics(32, “John Bury”);
        }
```

```
------------------------------------------------------------------

------------------------------------------------------------------
Mathematics.java
-------------------------------------------------------------------
public class Mathematics {
        int seatsAvailable;
        String teacherName;

        public Mathematics(int seats, String name) {
                seatsAvailable = seats;
                teacherName = name;
        }
}
------------------------------------------------------------------
```

What we can see in the example is a school class with its main method and an object called mathClassroom. An object as you might guess is created by typing the class name followed by what name you want that object to have and then the ‘new’ keyword followed by the class name. After the class name we have some parameters passed in the brackets. Are these the suppliers for the variables?

Think of the syntax above as some kind of method call. In the left side we have the class name followed by the object name that we choose. In the right hand side we have the keyword ‘new’ that creates that object and then the constructor method is provided with arguments. The arguments must be provided in the order of the parameters declared in the constructor method.

```
public Mathematics(int seats, String name)
Mathematics(32, “John Bury”)
```

So in the end we create an object called mathClassroom in the class School that has 32 Seats available and a teacher called John Bury. How can we display the results of the instance methods in the class Mathematics?

Remember! In the example above we have JUST created an object, nothing else.

```
-----------------------------------------------------------------
School.java
-----------------------------------------------------------------
public class School {
        public static void main(String args[]) {
        Mathematics mathClassroom = new Mathematics(32, “John Bury”);

        mathClassroom.getSeatsAvailable();
        mathClassroom.getTeacherName();
        }
}
-----------------------------------------------------------------
```

```
-----------------------------------------------------------------
OUTPUT
-----------------------------------------------------------------
The number of seats available in the Mathematics classroom is 32
The mathematics teacher is called John Bury
```

In the above example we can see that in order to access/call the instance methods from a class, first we need to reference using a ‘.’. We take the object name then reference it using the dot and then type the instance method’s name followed by the parenthesis that can/not have arguments passed based on what type of method is it - accessor or mutator.

Now that we know how to create/populate a class, create an object and use it we will talk about mutator and accessor methods.

What is an accessor method?

In the example above all the instance methods were accessor. Those types of methods are used to retrieve/return an information. We were able to successfully retrieve the number of seats available in the classroom and the teacher's name.

The accessor methods are distinguished by the 'get' keyword used in front of the name given to the method.

```
public void getTeacherName()
public void getSeatsAvailable()
```

What is a mutator method?

A mutator method is used to set a value to a private variable field. The mutator method is having a naming convention scheme that make use of the keyword 'set' that is the prefix of the method name.

Let's complete the code example:

Mathematics.java

```
public class Mathematics {
        private int seatsAvailable;

        public Mathematics(int seats) {
                seatsAvailable = seats;
        }

        public void setSeatsAvailable(int seatsNumber) {
                seatsAvailable = seatsNumber;
```

```
	}

	public void getSeatsAvailable()
	{
		System.out.println(“The number of seats available in the Mathematics classroom is ” + seatsAvailable);
	}
}
```

Here we create the mutator method that will update the private variable ‘seatsAvailable’

School.java

```
public class School {
	public static void main(String args[]) {
		Mathematics mathClassroom = new Mathematics(32);

		mathClassroom.setSeatsAvailable(45);
		mathClassroom.getSeatsAvailable();
	}
}
```

OUTPUT

```
The number of seats available in the Mathematics classroom is 45
```

We created the object mathClassroom with 32 initial seats available and then we ‘overwrite’ the value by using the ‘setSeatsAvailable’ mutator method into 45 available seats.

Multiple use of an outside class

In all the examples above we have learned how a class is created, its elements and how to define an object, just one! In order to understand the power of the classes and the objects then we will create more objects in a class by making use of the diagram used above.

First example will use 2 classes, the Mathematics Class and the School Class. The School class will make use of the Mathematics class more than just once.

```
-----------------------------------------------------------------------------
Mathematics.java
-----------------------------------------------------------------------------
public class Mathematics {
    // Part 1
    private int seatNumber;
    private String personName;
    private int penNumber;
    private boolean lateFlag;

    // Part 2
    public Mathematics(String name, int seat, int pen, boolean late) {
        personName = name;
        seatNumber = seat;
        penNumber = pen;
        lateFlag = late;
    }

    // Part 3
    public void studentInfo()
    {
        System.out.println(“This is ” + personName);
        System.out.println(“His/Her seat number is: ” + seatNumber);
        System.out.println(“He/She got: ” + penNumber + “ pens”);
        System.out.println(“It is ” + lateFlag + “ that he/she is late”);
    }
}
----------------------------------------------------------------------------
```

In our example there are 3 parts:

The first part defines 4 instance variables that will retain the student information.

In the second part we have a constructor method that will be the middle between the School class and the Mathematics class varible which will assign whatever value you set in the School class to each instance variable in the Mathematics Class.

The third part has a method of displaying the information about each student. Once a student is created in the School class we can call this method that will give us some details about his/her mathematics class.

```
--------------------------------------------------------------
School.java
--------------------------------------------------------------
public class School {
    public static void main(String args[]) {
        // Create them
        Mathematics person1= new Mathematics(“John”, 33, 2, true);
        Mathematics person2= new Mathematics(“Mark”, 24, 1, false);
        Mathematics person3= new Mathematics(“Felix”, 12, 5, true);
        Mathematics person4= new Mathematics(“Jess”, 2, 3, false);

        // Introduce them
        person1.studentInfo();
        person2.studentInfo();
        person3.studentInfo();
        person4.studentInfo();
    }
}
```

Above we have the School class in which we have created 4 persons which are students in the school and each of them belong in the mathematics classroom. Each of them has a name, a seat number, a number of pens and some are late some are not.

If we run this School class the output will be:

```
This is John
His/Her seat number is: 33
He/She got: 2 pens
It is true that he/she is late
This is Mark
His/Her seat number is: 24
He/She got: 1 pens
It is false that he/she is late
This is Felix
His/Her seat number is: 12
He/She got: 5 pens
It is true that he/she is late
This is Jess
His/Her seat number is: 2
He/She got: 3 pens
It is false that he/she is late
```

Basically we make use of the mathematics class to create 4 people with different specs each. They are all taking the mathematics course and they are all created using a constructor method that receives each person's specs and assign them to the instance variables.

Remember!

The constructor is able to do the same thing as the mutator method but the reason we use the method is because it is easy to change the value afterwards. If we use a constructor to assign values then we want these values to be changed then we can make use of mutator methods to change the values.

Also pay attention at the fact that the ‘studentInfo’ method is just an accessor method since it just takes the values passed in the School Class and retrieve them in sentences in order to have a meaning.

Modifier Types

After finishing with the basics of classes and objects we can move on with understanding certain keywords that appear in the text which were not explained before.

There are 2 types of modifiers in Java: Access and Non Access Modifiers.

Access Modifiers

Public

A class, method, variable, constructor, etc. that is declared as being public will be able to be accessed from any other class.

For example, the main method has to be public all the time since it is the most important component of a program that has to be accessed by the other classes interfaces, etc.

```
public static void main(String args[]){}
```

Important!
No keyword in front of the component refers to the fact that they are by default declared as being public.

Private

When we put the ‘private’ keyword in front of a method, variable etc. then that method, variable or component can only be accessed within the class it is defined.

Important!
Interfaces and classes cannot be private.
The only way we can access components outside the class they are defined is by using the accessor and mutator methods which are declared as being public.

Example from above:

```
public class Mathematics {
	private int seatsAvailable;

	public Mathematics(int seats) {
		seatsAvailable = seats;
	}

	public void setSeatsAvailable(int seatsNumber) {
		seatsAvailable = seatsNumber;
	}

	public void getSeatsAvailable()
	{
		System.out.println(“The number of seats available in the 
Mathematics classroom is ” + seatsAvailable);
	}
}
```

In the code above we can see that the instance variable is declared as being private. The getters and setters are declared public so the outside class can get access to the variable by using the getters and setters.

Practice: Make a complex student system for different classrooms

Inner classes

After learning about how to create a class, make use of it and create different objects we will look at how to have multiple classes inside one class and how to make instances of them.

We will use the School example above.

In the example above we were having a class for each classroom and all the classes(Mathematics, Biology, etc.) were linked in the School class which was the main class that has them all.

Now we will use the School Class as an outer-class and all the other classes that were linked to it will be inner classes. Check the example below:

```
public class School{
        public class Mathematics{
                public void getName(){
                System.out.println(“This is the Mathematics class”);
        }
        }
        public class Biology{
                public void getName(){

                System.out.println(“This is the Biology class”);
        }
        }
        public class Spanish{
                public void getName(){
                System.out.println(“This is the Spanish class”);
```

```
    }
    }
    public class Physics{
        public void getName(){
        System.out.println(“This is the Physics class”);
    }
    }
}
```

Here we can see that there is the School Class that has all the other Classes inside it, each having a print statement that specifies the class name.

Now let’s create a new class called ‘MyClassroom’ that will create instances of what classes a student is attending.

```
public class MyClassroom{
    public static void main(String args[]){
        School school = new School();

        School.Spanish spanish = school.new Spanish();
        spanish.getName();
}
}
```

Here we make an instance of the outer class which is School Class and then we make an instance of the inner class which is Spanish Class. The Spanish Class has a getter which prints the class name. We use the instance to call the getter afterwards.

OUTPUT

```
This is the Spanish class
```

Method Local Inner Class

Beside having all the classes in one class we can also have classes inside methods. For the example above we grouped all the course classes into one class called School.

In school we can have a method for example called totalCourses that will retrieve the number of classes that are in the School. For our example above there are 4 classes. Let’s see:

```
public class School {
        public void totalCourses() {
                public class Print{
                        public void printTotalNumber() {
                                System.out.println(“4 Classes”);
                        }
                }
        }

        public class Mathematics{
                …..
        }

        public class Biology{
                …..
        }
        …...
        Print printer = new Printer();
        printer.printTotalNumber();
}
```

As you see here we have the outer class ‘School’, then inside is a method called ‘totalCourses’. In the ‘totalCourses’ method there is a class created called ‘Print’ with a method called printTotalNumber that prints the number of classes inside the School Class.

In order to make it more simple to execute the totalCourses method, we make an instance of the Print class before closing the outer class bracket

Now let’s take a look at the main method inside our MyClassroom class:

```
public class MyClassroom{
        public static void main(String args[]){
                School school = new School();
                school.totalCourses();
        }
}
```

In the main method we only make an instance of the outer class and then we call the method and the final **output** will be:

```
4 Classes
```

Static Nested Class

The difference with the static nested class is the fact that you do not need to make an instance of the outer class. Let’s take the examples from the inner class beginning.

```
public class MyClassroom{
        public static void main(String args[]){
                School school = new School();

                School.Spanish spanish = school.new Spanish();
                spanish.getName();
}
}
```

Here is the example when we made an instance of the outer class ‘School’. If we adapt it for the static nested class then we get:

```
public class MyClassroom{
        public static void main(String args[]){

        School.Spanish classroom = new School.Spanish();
        classroom.getName();
        }
}
```

Inheritance

Inheritance is the process in which a class acquires the properties(methods, etc.) of another class.

The class which inherits the properties is called **Subclass** and the class that has the properties inherited is called **Superclass**.

Let’s say we have two classrooms: Mathematics and Biology. We want to have the same seats in the Biology classroom as there are in the Mathematics class so we need to inherit the properties of the seats(Brand, color, size, etc.). How do we do that?

```
public class Mathematics{}
public class Biology{}
```

extends Keyword

In order to inherit the properties of the Mathematics class we use the following syntax:

```
public class Mathematics{}
public class Biology extends Mathematics{}
```

PRACTICE

You are going to have a quizz to test your understanding so far! Good Luck!

1. Select the right definition of a class called “Human”:
    a. public class main Humans
    b. class main Human
    c. public class Humans
    d. public static Human
    e. public class Human

2. What is a class for object-creating?
    a. A template for creating another class
    b. A class is a template for creating classrooms for the school
    c. A class is an object used in school
    d. A template for creating different objects which defines its properties and behaviors
    e. A blueprint for creating Mathematics class

3. Complete the following constructor method:

```
-----------------------------------------------------------------
Phone.java
------------------------------------------------------------------
public class Phone {
        int megapixels;
        String phoneName;

        public Phone(int mPixels, String pName) {
                ……?……...
        }
}
----------------------------------------------------------------
    a.  Megapixels = mPixels;
        phoneName = pName;

   b. mPixels = megapixels;
      pName = mPixels;
   c. pName = mPixels;
     Megapixels = pName;
   d. megapixels = mPixels;
     phoneName = pName;
  e. megapixels = pName;
     PhoneName = pName;
```

4. Create an object called tea that will output:

        It takes 3 minutes to make a tea
        I like the green tea.

```
----------------------------------------------------------------
Tea.java
-----------------------------------------------------------------
public class TeaBags {
        int minutesToMake;
        String teaType;

        public TeaBags(int min, String type) {
                minutesToMake = min;
                teaType = type;
        }

        public void info() {
          System.out.println(“It takes “ + minutesToMake + “ minutes to make
a tea”);
          System.out.println(“I like the “ + teaType + “ tea.”);
         }
}
-----------------------------------------------------------------
```

a.

b. mPixels = megapixels;
pName = mPixels;

c. pName = mPixels;
Megapixels = pName;

d. megapixels = mPixels;
phoneName = pName;

e. megapixels = pName;
PhoneName = pName;

Appendix B

The contents of this appendix refer to the full research paper conducted on the 49 most popular online learning resources. For the purpose of the table size, the paper is oriented in the landscape mode.

Popular eLearning Websites Comparison

The table below is comparing the most popular and known eLearning resources/websites with a pro/cons approach. At the end of the table, there is the "Java Just in Time" Book that has its up/downs. The book can be compared with the websites and there we can see that there is no interaction or responsiveness between the book and the user which is totally normal since it is a book. The good thing about the book is that it takes the beginner user and teach him every Java aspect, from a program simple as hello world and what is Hardware/Software to Generics and Data structures which makes it a great resource in terms of structure.

By analysing 49 websites there are problems with most of them, and one of these problems is interactivity and the responsive design which most of them lack. Other problems that appear in the websites with great user experience is the lack of content in the matter of java advanced topics since most of the content is pretty basic. Although the websites with great resources cost money as well, from 25$ to 249£ which is a lot. The difficulty is easy to hard for most of them but mostly easy because of the lack of content. Most of them have a theory part but is either poor or a bunch of theory and documentation that will not outline the important parts of the java chapters presented. The practice content is mostly available but there is a significative number of websites that have hard code challenges or no real-time feedback at all, and there are websites with no practice content at all.

The aim of this project is to create a website where there will be responsive web design implemented and interactivity between the platform and the user, with quality and outlining content for every type of users, quizzes and practice exercises for the purpose of remembering and applying the theory, with a easy to hard approach in terms of difficulty level and all the content being free for students and other type of users.

Theory Content

In matter of theory content, the website should be able to provide enough information about a certain concept and have a structure that will make the user understand terms and techniques from the simplest to the hardest ones. A good course will explain these terms in a way that the user will not feel overwhelmed or bored.

Practice Content

The practice content refers to the amount of problems/exercises a user receives from a website so that the theory will be together with practice. By combining theory and practice, the user will be able to understand and have a better idea of the usability of the terms and techniques in real world. If the user will practice coding intensively then he/she will get better and use the techniques and terms better than just reading them. The main idea is that the user will see and try to solve different problems and situations that are not possible in theory since the number of problems and their solving is infinite.

Interactive design

When we talk about interactive design we refer to the fact that the website has a user friendly design, easy-to-use and made in such a way that even the inexperienced user will know how to use it and choose a path or a course. The user might receive trophies based on how much work and how much content he had completed. Examples of trophies might be even if the user is going on the website daily or at a certain interval, or even if the user is doing a daily challenge he/she might receive appreciation.

Difficulty

A **Novice Programmer** will need guidance and will start with the basics. They probably won't have had any previous experience or could be new to programming and as well struggle with some elements of the language. Without guidance they could struggle in understanding how java and programming works.

A **Competent Programmer** will have had experience, probably more than 2-3 years in the field and they might be able to tackle some tasks and most of them on their own. Sometimes they might need help or an idea of how to approach a certain problem.

An **Advanced Programmer** will have a significant amount of experience usually in excess of 5+ years. They will be confident in their daily work and be able to design solutions as well as easily completing tasks. They will almost certainly have leadership skills and be able to mentor the junior developers.

Difficulty Level	**Topics**
Novice	● **Basic Syntax** ● **Basic Datatypes** ● **Numbers** ● **Characters** ● **Strings** ● **Variable Types** ● **Basic Operators** ● **Decision Making** ● **Loops** ● **Arrays** ● **Date & Time** ● **Methods** ● **Files and I/O** ● **Exceptions** ● **Documentation**

Competent	- **Object & Classes** - **Modifier Types** - **Inner classes** - **Inheritance** - **Overriding** - **Polymorphism** - **Abstraction** - **Encapsulation** - **Collections**
Advanced	- **Interfaces** - **Packages** - **Data Structures** - **Generics** - **Serialization** - **Networking** - **Multithreading** - **Applet Basics**

Table Grading

Interactive Design
0(No interactive content/static page) -->10(Great user experience - interactive content)

Practice Content
0(No quiz/Code challenge) -->10(Lot of Code challenges + Quizzes)

Theory Content
0(No theory at all) --> 10 (Full Java documentation)

	Interactive Design	Practice Content	Theory Content	Difficulty	Price	Pros	Cons
Androidauthority	1	0	3	Novice	Free	Easy and fast to read if you just need to know what is java but nothing more	No detail, no depth, no interactiveness, just a popular article
Beginnersbook	1	0	9	Novice - Competent	Free	Very good for its theory and pictures	Bad because there is no code challenge quiz or trophies or interactiveness that will drive the reader to use it more often

Best Programming Language For Me	1	0	1	Novice	Free	Explains why java is good and sends you to links categorised by the kind of resource it is(article, website)	Nothing on its own, it just sends you to other programming websites(popular ones)
Code.tutsplus	0	0	3	Novice	Free	No point, there are way better articles	It is popular but there is no practice content no interactivity
Codeavengers	7	9	7	Novice	Free	Very good for website development beginners	No Java tutorial even though it is very popular
Codecademy	9	10	6	Novice	Free/Paid	Interactive design that motivates you to come back	The bad thing is that it is simple, extremely simple especially in terms of java
Codechef	2	0	0	Novice	Free	Maybe it is good for the reference it has	Bad design, no original content, no code challenge or quiz
Codefights	10	10	0	Novice-Advanced	Free	Very nice looking with good programming challenges	No java theory or java based course

Codeforces	**1**	**10**	**0**	**Advanced**	**Free**	**Good for pros**	**No interactive content, no theory nothing more than just an ugly challenge website**
CodeMentor	**0**	**0**	**0**	**Novice**	**Free**	**Good as reference website**	**Nothing original, just references**
Coderbyte	**7**	**10**	**1**	**Novice**	**Paid**	**Code challenges and videos**	**Poor website and content that have potential to be better but by now there are better alternatives**
Codeschool	**8**	**10**	**8**	**Novice**	**Paid**	**Interactive design and quizzes**	**No Java tutorial even if it is a popular and good website**

Codewars	9	10	2	Novice-Advanced	Free	Good for improving programming skills but it is mostly for advanced users	In order to have an account you need to complete a code challenge in a programming language of your choice which is not for beginners since this project assumes that beginners with no programming experience will use a website to learn some
CodingBat	1	9	0	Novice-Advanced	Free	Good for practice	No java course just challenges and no interactive content
Computer Science Online	0	0	3	Novice	Free	Good for facts checking and getting reference	No java tutorial at all just references

Coursera	4	6	8	Novice-Advanced	Free/Paid	Variety of courses and the universities such as Stanford	Bad because the course content might be released weekly and the user needs to apply for a course which might start in a month or so, without having the choice of hovering over the course content and go to the topic he/she wants to
Edureka	2	4	7	Novice	Paid	A full java course	249£ to enrol and not that friendly platform
Edx	3	6	8	Novice-Advanced	Free/Paid	Diversity of courses to pick from	The courses are not interactive and the user might get bored fast.
Freecodecamp	0	8	7	Novice	Free	Web Development Bootcamp type	There is no java course
Hackerearth	7	9	6	Novice	Free	Good for beginners	No Java content

Hackerrank	**8**	**9**	**5**	**Competent - Advanced**	**Free**	**A lot of code challenges and even job opportunities**	**The website is not for beginners and it might be hard for some people, even if you are pro**
Hackr	**0**	**0**	**0**	**Novice**	**Free**	**Nothing**	**Just an empty forum that claims that it can find you a java buddy to learn with**
Homeandlearn	**0**	**0**	**7**	**Novice**	**Free**	**Good java documentation and examples**	**No interactive content or practice content**
IBM	**0**	**2**	**8**	**Novice**	**Free**	**Good examples for the java topics**	**No code challenge, weak quizzes, poor structured and represented content. A potentially good tutorial that nobody cared about.**
Introcs.cs.princeton	**0**	**0**	**0**	**Novice**	**Free**	**Code for different programs in java**	**No tutorial at all**
Javatpoint	**0**	**0**	**8**	**Novice**	**Free**	**Integrated compiler that shows how the code runs**	**No practice content and poor interactive content**

JavaWorld	0	0	8	Novice	Free	Good for revise	Bad because of the format, no code challenge or quizzes, no interactive content
KhanAcademy	8	7	7	Novice	Free	Good for beginners - Intro to computing	No java course
Learn Java Online	3	6	5	Competent	Free	It might be handy for some people	The big problem is the missing content, once you click on some chapters(Links) they will redirect you to a 500 error code
Learnxinyminutes	2	9	0	Novice-Advanced	Free	Good for practice	No java course or theory just a link to the Oracle documentation
Leetcode	2	9	0	Novice-Advanced	Free	Good for practice	Nothing involving a java course just some code challenges
Lynda	4	0	8	Novice-Advanced	Paid	It might help if you are interested in a certain java use but not a java basic to	No Java from easy to advanced tutorial with quizzes and/or code

						advanced website	challenges integrated
Mkyong	0	0	6	Novice - Competent	Free	If you need to check some certain things like stackoverflow	No tutorial or structure at all
NetBeans	0	0	2	Novice	Free	Nothing	I have no idea why this website exists, it is just a bunch of external links that are not useful at all
MIT Courseware	3	6	6	Novice	Free	It is good because of the simplicity and the examples given	No real-time feedback or challenge or quiz, no responsive web design at all
Oracle	0	0	10	Novice-Advanced	Free	It has all the Java concepts out there so it is best for checking a technique or keyword that you might not know or identify in a code	It is so detailed that a link will lead you to a list of links related to that and one link from that list will link you to a new list of links, basically making it pretty impossible to learn from.

Programiz	0	0	6	Novice	Free	Easy but poor and incomplete java tutorial	Incomplete and too easy, better alternatives
Project Euler	0	6	0	Advanced	Free	Good for improving the skills you already have and challenge yourself	Not a tutorial for beginners and most of its content is for pro users
Reddit	0	3	7	Novice-Advanced	Free	Good when in need of a certain answer for a certain situation	It has information and answers but not ordered(no path)
Sololearn	8	7	9	Novice	Free	Probably the best in this list because it is interactive, it has medium quantity and quality theory and quizzes	There can be more content and the quizzes can be longer and with more questions not just three or four per each chapter
Spoj	2	9	0	Novice-Advanced	Free	Good to practice programming	No theory for java

TeamTreehouse	10	7	7	Novice	Paid	It is very good but the Java content is not that much. Most of that courses are explaining the uses of Java in creating applications rather than teaching just Java.	There can be more content for Java as programming language rather than its applications and beside that there can be also more added code challenges and quizzes for each section.
Tech Gig	8	8	3	Novice	Free	Good to practice programming	No theory for Java just challenges
Tutorialspoint	2	0	9	Novice -Advanced	Free	Good when you need a term or technique explained but not for practicing	It should have interactiveness to motivate the reader and it should also have code challenges or quizzes.
Udacity	8	8	8	Novice-Advanced	Free/Paid	Easy to understand and learn new things that are happening	No Java from easy to advanced course, just an intro and then the java uses, similar to Teamtreehouse

Udemy	5	5	8	Novice	Free/Paid	Wide variety of courses	It does not provide a unique and great course, mostly a course advertisement website
Vogella	0	0	2	Novice - Advanced	Free	Covers most java terms and details	Just a website with a bunch of theory about java but no code challenges at all or even a quiz or so
Java Just in Time Book	0	9	10	Novice-Advanced	Paid	Covers the Java topics from HelloWorld to Generics and other advanced topics with good coursework practice	Sometimes involves code to be completed that is only available to UoM students. There is no E-learning in there, just the book and its normal format.

Table 1: Graded table based on the above marking scheme

	Interactive Design	Practice Content	Theory Content	Difficulty	Price	Pros	Cons
Androidauthority	Just a popular article on a website	Nothing to practice just some advice on how to run the code(theory)	An article with theory content but facile, just basic definitions and some examples	Easy	Free	Easy and fast to read if you just need to know what is java but nothing more	No detail, no depth, no interactiveness, just a popular article
Beginnersbook	No interactive or responsive Than just some pictures	No practice content	Very good and strong website in terms of theory content, from simple java to advanced topics	Easy - Medium - Hard	Free	Very good for its theory and pictures	Bad because there is no code challenge quiz or trophies or interactiveness that will drive the reader to use it more often
Best Programming Language For Me	No interactiveness just some infographics	No practice content	Explains why java is good to learn but nothing more and then sends you to some links	Easy	Free	Explains why java is good and sends you to links categorised by the kind of resource it is(article, website)	Nothing on its own, it just sends you to other programming websites(popular ones)

Code.tutsplus	**No interactive content**	**No content for practice**	**Poor theory just an article basically that tries to cover Java language in a small web page**	**Easy**	**Free**	**No point, there are way better articles**	**It is popular but there is no practice content no interactivity**
Codeavengers	**Interactive website that even use a tool similar to Dreamweaver from Adobe in the website**	**Practice with code challenge**	**Videos that explain the definitions and uses**	**Easy**	**Free**	**Very good for website development beginners**	**No Java tutorial even though it is very popular**
Codecademy	**Interactive with trophies and popup boxes responsive**	**It is basically a compiler and you just type the code**	**Easy content not that hard or advanced at all**	**Easy**	**Free/20$**	**Interactive design that motivates you to come back**	**The bad thing is that it is simple, extremely simple especially in terms of java**
Codechef	**Bad layout and no interactive content, just a bunch of boxes and details thrown on the page**	**No code challenge or practice involved**	**No theory involved just links to other websites that teach programming**	**Easy**	**Free**	**Maybe it is good for the reference it has**	**Bad design, no original content, no code challenge or quiz**
Codefights	**Beautiful design and a lot of interactive content, just beautiful**	**Code challenge based**	**No theory just practice for coding**	**Easy-Hard**	**Free**	**Very nice looking with good programming challenges**	**No java theory or java based course**

Codeforces	Plain html with some css	Just challenges	No theory	Hard	Free	Good for pros	No interactive content, no theory nothing more than just an ugly challenge website
CodeMentor	No interactive content	No challenges	No theory just reference	Easy	Free	Good as reference website	Nothing original, just references
Coderbyte	Pretty interactive but not as good as teamtreehouse	Code challenges like Codewars	Poor theory presented in videos	Easy	Paid	Code challenges and videos	Poor website and content that have potential to be better but by now there are better alternatives
Codeschool	Interactive website with responsive web design and friendly interface	Practice by completing quizzes or code challenges, similar to Teamtreehouse	Enough theory to complete the code challenges but not complicated or advanced topics	Easy	20$/month	Interactive design and quizzes but no Java tutorial	No Java tutorial even if it is a popular and good website

Codewars	Interactive and interesting website with trophies and achievements	Practice by completing code challenges	Not enough theory, rather made for medium to pro users	Hard	Free	Good for improving programming skills but it is mostly for pro users	In order to have an account you need to complete a code challenge in a programming language of your choice which is not for beginners since this project assumes that beginners with no programming experience will use a website to learn some
CodingBat	Plain html with some css	Practice by code challenge	No theory	Easy-Hard	Free	Good for practice	No java course just challenges and no interactive content
Computer Science Online	No interactive content	No practice content	A PRO reference based website with facts	Easy	Free	Good for facts checking and getting reference	No java tutorial at all just references

Coursera	Not a responsive design website, but a normal one with courses.	The practice part is either by submitting homework or assignments or in form of quizzes but that is different based on the course you pick	Depends on the university or the individual you picked the course from.	Easy-Medium-Hard	Free or 50$ + Certificate	Good because of the variety of courses and the universities such as Stanford that teach them	Bad because the course content might be released weekly and the user needs to apply for a course which might start in a month or so, without having the choice of hovering over the course content and go to the topic he/she wants to
Ereduka	Little interactive content - poor	Assignments that each user should do them in his own environment	Good java theory with not that good structure	Easy	Expensive	A full java course	249£ to enrol and not that friendly platform
Edx	Interactive based on who teach that content - similar to Coursera	Practice content is mostly quizzes and code challenge that should be submitted as a file - no real-time checker	Theory different based on what course you pick	Easy-Hard	Free/Paid	Diversity of courses to pick from	The courses are not interactive and the user might get bored fast.
Freecodecamp	No interactive content	The website is code challenge based	Enough theory to learn something	Easy	Free	Web Development Bootcamp type	There is no java course

Hackerearth	Interactive content	The website is code challenge based	There is basic learning content	Easy	Free	Good for beginners	No Java content
Hackerrank	Interactive and responsive somehow	Practice by completing code challenges, similar to Code Wars	Theory enough to complete the code challenge but most of the content is for Novice - Pro	Medium-Hard	Free	A lot of code challenges and even job opportunities	The website is not for beginners and it might be hard for some people, even if you are pro
Hackr	No interactive content	No practice content	No Java theory, just an empty forum	Easy	Free	Nothing	Just an empty forum that claims that it can find you a java buddy to learn with
Homeandlearn	No interactive content	No practice content	Good java theory	Easy	Free	Good java documentation and examples	No interactive content or practice content
IBM	No interactive web design	A quiz at each end of a chapter that is not even a e-quiz, you should print it or put the answers on paper	Nice java theory but to formal	Easy	Free	Good examples for the java topics	No code challenge, weak quizzes, poor structured and represented content. A potentially good tutorial that nobody cared about.
Introcs.cs.princeton	No interactive content	No practice content	No Java theory, just a code repository	Easy	Free	Code for different programs in java	No tutorial at all

Javatpoint	No interactive content	No practice content	Good java theory	Easy	Free	Integrated compiler that shows how the code runs	No practice content and poor interactive content
JavaWorld	No interactive content, just article based	No code challenge	Theory in form of articles	Easy	Free	Good for quick checking	Bad because of the format, no code challenge or quizzes, no interactive content
KhanAcademy	Interactive and responsive design with achievements and trophies	Practice by doing code challenges or quizzes	Videos with teachers that tell you about the content	Easy	Free	Good for beginners - intro to computing	No java course
Learn Java Online	It does not give you any achievement and it is no responsive web design involved	You have a compiler in which there is java code to be completed	The theory is medium not so in depth and not enough for the code challenges	Medium	Free	It might be handy for some people	The big problem is the missing content, once you click on some chapters(Links) they will redirect you to a 500 error code
Learnxinyminutes	No interactive content just a compiler	Code challenge based	No theory	Easy-Hard	Free	Good for practicing	No java course or theory just a link to the Oracle documentation
Leetcode	No interactive content, just some links with code challenges	Code challenge based	No theory	Easy-Hard	Free	Good for practice	Nothing involving a java course just some code challenges

Lynda	**Not that interactive website but a course list one**	**No code challenge or quiz**	**Videos with teachers that tell you about the content**	**Easy-Hard**	**Paid**	**It might help if you are interested in a certain java use but not a java basic to advanced website**	**No Java from easy to advanced tutorial with quizzes and/or code challenges integrated**
Mkyong	**No interactive content**	**No code challenge**	**Theory structure is a mess no easy to advanced topics, just random topics about java uses**	**Easy-Hard**	**Free**	**If you need to check some certain things like stackoverflow**	**No tutorial or structure at all**
NetBeans	**No interactive content**	**No code challenge**	**Horrible content, lack of a LOT of resources**	**Easy**	**Free**	**Nothing**	**I have no idea why this website exists, it is just a bunch of external links that are not useful at all**
MIT Courseware	**Not that interactive website but a course list one with PDFs**	**Assignments that each user should do them in his own environment**	**Theory simple explained in PDF slides**	**Easy**	**Free**	**It is good because of the simplicity and the examples given**	**No real-time feedback or challenge or quiz, no responsive web design at all**

Oracle	**Not interactive at all**	**No code challenge or quizz**	**The source of Java documentation**	**Easy-Hard**	**Free**	**It has all the Java concepts out there so it is best for checking a technique or keyword that you might not know or identify in a code**	**It is so detailed that a link will lead you to a list of links related to that and one link from that list will link you to a new list of links, basically making it pretty impossible to learn from.**
Programiz	**No interactive content**	**No code challenge**	**Basic java course with some details and examples but nothing advanced**	**Easy**	**Free**	**Easy but poor and incomplete java tutorial**	**Incomplete and too easy, better alternatives**
Project Euler	**No interactive content**	**Basically a website with code challenge but no website compiler, you do it on your environment**	**No Java theory, just challenges**	**Hard**	**Free**	**Good for improving the skills you already have and challenge yourself**	**Not a tutorial for beginners and most of its content is for pro users**
Reddit	**No interactive web design**	**No code challenge, maybe just tasks to complete on your own environment**	**A forum similar to Stackoverflow**	**Easy-Hard**	**Free**	**Good when in need of a certain answer for a certain situation**	

Sololearn	It is interactive with trophies a certificate for each course and responsive based on your entries	Code that needs to be completed after each part of theory	Structured in chapters and there is enough theory to know java at a medium to advanced level	Easy	Free	Probably the best in this list because it is interactive, it has medium quantity and quality theory and quizzes	There can be more content and the quizzes can be longer and with more questions not just three or four per each chapter
Spoj	Not that much interactivity	Code challenge based	Does not provide theory just code challenges	Easy-Hard	Free	Good to practice programming	No theory for java
TeamTreehouse	Very interactive and responsive design from trophies earned to entry responses	Quizzes and code challenges for each chapter or section of a course	Structured in paths or courses the website has videos with instructors that will explain the concepts.	Easy	25$/50$ +7 Day Trial	It is very good but the Java content is not that much. Most of that courses are explaining the uses of Java in creating applications rather than teaching just Java.	There can be more content for Java as programming language rather than its applications and beside that there can be also more added code challenges and quizzes for each section.
Tech Gig	Interactive content with achievements	Similar to Hackerrank - code challenge based	No theory just challenges even if the title says “Java Essentials”	Easy	Free	Good to practice programming	No theory for Java just challenges
Tutorialspoint	Normal website with links	No quiz or code challenge	Java theory will reach advanced levels but it is like the beginnersbook example	Novice-Advanced	Free	Good when you need a term or technique explained but not for practicing	It should have interactiveness to motivate the reader and it should also have

							code challenges or quizzes.
Udacity	**Website with interactive content that has achievements and different trophies**	**Quizzes and code challenge integrated in the website**	**Theory in form of videos - it has quality**	**Novice-Advanced**	**Free/Paid**	**Easy to understand and learn new things that are happening**	**No Java from easy to advanced course, just an intro and then the java uses, similar to Teamtreehouse**
Udemy	**Normal website with little interactive design**	**Some courses have quizzes in them and code challenges that are like Coursera**	**Each course is different based on what you pick and its instructor**	**Novice**	**Free/Paid**	**Wide variety of courses**	**It does not provide a unique and great course, mostly a course advertisement website**
Vogella	**No interactive content on the website**	**No practice parts**	**A bunch of text with some code examples but nothing more**	**Novice-Advanced**	**Free**	**Covers most java terms and details**	**Just a website with a bunch of theory about java but no code challenges at all or even a quiz or so**

Java Just in Time Book	-	Coursework that involves making programs for each chapter or subchapter of the book	Easy to Advanced theory, very good structure	Easy-Hard	£10	Covers the Java topics from HelloWorld to Generics and other advanced topics with good coursework practice	Sometimes involves code to be completed that is only available to UoM students. There is no E-learning in there, just the book and its normal format.

Table 2: Websites review based on the Table 1 ratings (Commented)

Sorting the websites based on grades and categories

	Interactive Design	Practice Content	Theory Content
0	Code.tutsplus CodeMentor ComputerScienceOnline Freecodecamp Hackr Homeandlearn IBM Introcs.cs.princeton	Vogella TutorialsPoint Programiz Oracle NetBeans Mkyong Lynda Javaworld	Codechef Codefights Codeforces CodeMentor CodingBat Hackr Introcs.cs.princeton Learnxinyminutes

	Javatpoint JavaWorld Mkyoung NetBeans Oracle Programiz Project Euler Reddit Vogella JavaJustInTime	Javatpoint Introcs.cs.princeton Homeandlearn Hackr ComputerScienceOnline CodeMentor Codechef Code.tutsplus BestProgrammingLanguageForMe Beginnersbook AndroidAuthority	Leetcode ProjectEuler Spoj
1	AndroidAuthority Beginnersbook BestProgrammingLanguageForMe Codeforces CodingBat	-	Coderbyte BestProgrammingLanguageForMe
2	TutorialsPoint Spoj Leetcode Learnxinyminutes Edureka Codechef	IBM	Codewars NetBeans Vogella
3	Edx LearnJavaOnline MITCourseware	Reddit	TechGig ComputerScienceOnline Code.tutsplus AndroidAuthority
4	Lynda	Edureka	-

	Coursera		
5	Udemy	Udemy	Hackerrank
6	-	Project Euler LearnJavaOnline Edx Coursera	Codecademy Hackerearth Mkyong MIT Courseware Programiz
7	Codeavengers Coderbyte Hackerearth	KhanAcademy Sololearn TeamTreehouse	TeamTreehouse Reddit KhanAcademy Homeandlearn Freecodecamp Edureka Codeavengers
8	Udacity TechGig Sololearn KhanAcademy Hackerrank Codeschool	Freecodecamp TechGig Udacity	Codeschool Coursera Edx IBM Javatpoint Javaworld Lynda Udacity Udemy

9	Codecademy Codewars	Codeavengers Codingbat HAckerearth Hackerrank Learnxinyminutes Leetcode Spoj JavaJustInTime	TutorialsPoint Sololearn Beginnersbook
10	Team Treehouse Codefights	Codecademy Codefights Codeforces Coderbyte Codeschool Codewars	Oracle JavaJustInTime

Table 3: Table 1 sorted based on grades and categories

	Interactive Design	Practice Content	Theory Content	Mean

Codeschool	8	10	8	8.6
Codecademy	9	10	6	8.3
Sololearn	8	7	9	8
TeamTreehouse	10	7	7	8
Udacity	8	8	8	8
Codeavengers	7	9	7	7.6
Hackerearth	7	9	6	7.3
Hackerrank	8	9	5	7.3
KhanAcademy	8	7	7	7.3
Codewars	9	10	2	7
Codefights	10	10	0	6.6
Tech Gig	8	8	3	6.3
Java Just in Time Book	0	9	10	6.3
Coderbyte	7	10	1	6
Coursera	4	6	8	6
Udemy	5	5	8	6

Edx	3	6	8	5.6
Freecodecamp	0	8	7	5
MIT Courseware	3	6	6	5
Learn Java Online	3	6	5	4.6
Edureka	2	4	7	4.3
Lynda	4	0	8	4
Codeforces	1	10	0	3.6
Learnxinyminutes	2	9	0	3.6
Leetcode	2	9	0	3.6
Spoj	2	9	0	3.6
Tutorialspoint	2	0	9	3.6
Beginnersbook	1	0	9	3.3
CodingBat	1	9	0	3.3
IBM	0	2	8	3.3
Oracle	0	0	10	3.3
Reddit	0	3	7	3.3

Javatpoint	**0**	**0**	**8**	**2.6**
JavaWorld	**0**	**0**	**8**	**2.6**
Homeandlearn	**0**	**0**	**7**	**2.3**
Mkyong	**0**	**0**	**6**	**2**
Programiz	**0**	**0**	**6**	**2**
Project Euler	**0**	**6**	**0**	**2**
Androidauthority	**1**	**0**	**3**	**1.3**
Code.tutsplus	**0**	**0**	**3**	**1**
Computer Science Online	**0**	**0**	**3**	**1**
Best Programming Language For Me	**1**	**0**	**1**	**0.6**
Codechef	**2**	**0**	**0**	**0.6**
NetBeans	**0**	**0**	**2**	**0.6**
Vogella	**0**	**0**	**2**	**0.6**
CodeMentor	**0**	**0**	**0**	**0**
Hackr	**0**	**0**	**0**	**0**

Introcs.cs.princeton	**0**	**0**	**0**	**0**

*Table 4: Websites sorted by the best score**

**Best score refers to the mean between interactive content, theory content and practice content grades*