

# Software Design & Development

## CFS2160

Week 17 – Java Documentation

# Session Plan

- Discuss the importance of the Java Documentation.
- Take a look at the Java Documentation.
- Libraries
- A few questions about it
- Finally.

# Java Libraries

We can add extra functionality to our programme by importing additional libraries into our code. This allows us to use pre-existing classes to complete common and repeating tasks.

The majority of what we want to do has, most likely been done before. There is no value in creating code for these common tasks as it will probably be already within Java.

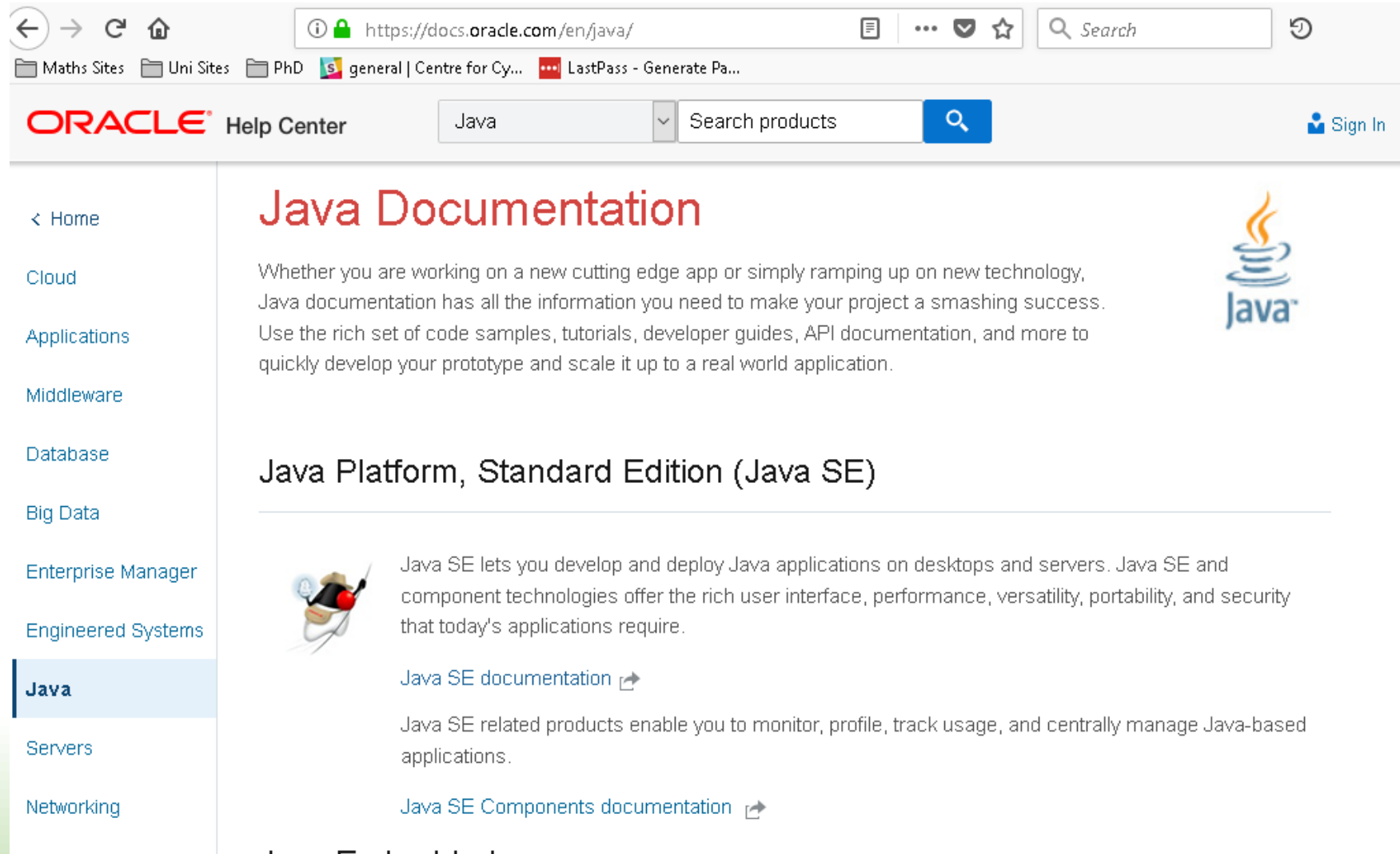
As programmers we need to learn the following:

1. It has been done before?
2. How do we use it?

If a library exists with the functionality we need, we do not need to know what it does or how it works, just how we can use it to our advantage!

We then add code to our programme that stitches the existing Java classes together to get the functionality we require.

# Java Documentation home page



The screenshot shows a web browser window with the URL `https://docs.oracle.com/en/java/`. The browser's address bar includes navigation icons, a search bar, and a list of open tabs. Below the browser window, the Oracle Help Center header is visible, featuring the Oracle logo, the text "Help Center", a dropdown menu set to "Java", a "Search products" input field, and a "Sign In" link. The main content area is titled "Java Documentation" and includes a paragraph about the documentation's purpose. A sidebar on the left lists various categories, with "Java" highlighted. The "Java Platform, Standard Edition (Java SE)" section is expanded, showing a description of Java SE and links to "Java SE documentation" and "Java SE Components documentation".

ORACLE<sup>®</sup> Help Center

Java Search products Sign In

< Home

Cloud

Applications

Middleware

Database

Big Data

Enterprise Manager

Engineered Systems


**Java**

Servers


Networking

## Java Documentation

Whether you are working on a new cutting edge app or simply ramping up on new technology, Java documentation has all the information you need to make your project a smashing success. Use the rich set of code samples, tutorials, developer guides, API documentation, and more to quickly develop your prototype and scale it up to a real world application.



### Java Platform, Standard Edition (Java SE)



Java SE lets you develop and deploy Java applications on desktops and servers. Java SE and component technologies offer the rich user interface, performance, versatility, portability, and security that today's applications require.

[Java SE documentation](#)

Java SE related products enable you to monitor, profile, track usage, and centrally manage Java-based applications.

[Java SE Components documentation](#)

Java Embedded

# Java Documentation

Handily, Java has an in depth set of documents on its website to support a programmer in their work.

We are currently using Java version 8 so lets start there.

<https://docs.oracle.com/en/java/>

Search for ArrayList (version 8) as it is something we already know about.

<https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>

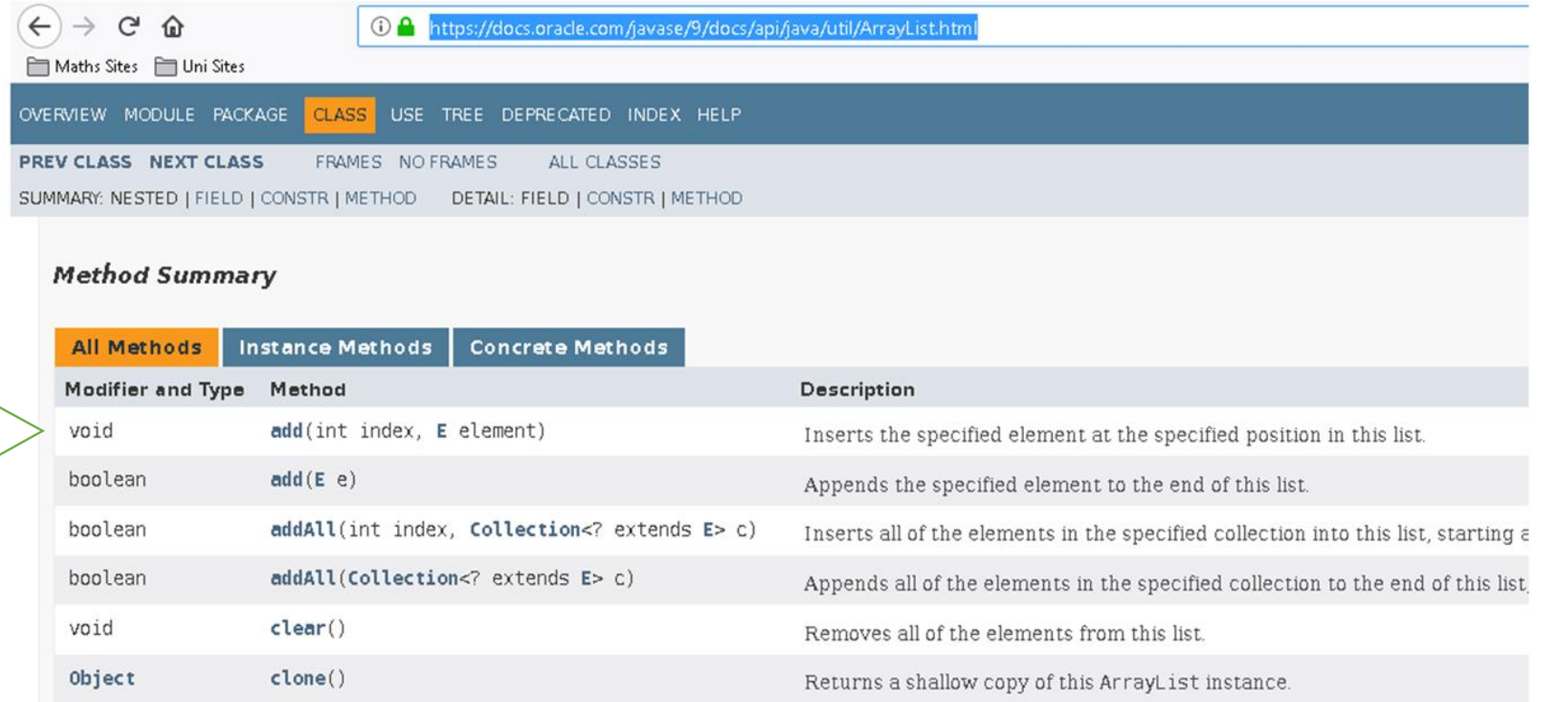
The documentation tells us a load of information about the ArrayList class, in particular its constructor and its methods.

# Looking At ArrayList

We can view information about all the methods available in the ArrayList package, as Tony says, “We do not need to know how it works, just how to use it!”

Method summary tells us all the method available to use in ArrayList.

You have used the .add() method many times before. The documentation tells us what it does, what values we can pass and what the method returns.



The screenshot shows the Java API documentation for the ArrayList class. The browser address bar displays the URL: <https://docs.oracle.com/javase/9/docs/api/java/util/ArrayList.html>. The navigation bar includes tabs for OVERVIEW, MODULE, PACKAGE, CLASS (selected), USE, TREE, DEPRECATED, INDEX, and HELP. Below the navigation bar, there are links for PREV CLASS, NEXT CLASS, FRAMES, NO FRAMES, and ALL CLASSES. The main content area is titled "Method Summary" and features three tabs: All Methods (selected), Instance Methods, and Concrete Methods. A table lists the methods with their modifiers, types, and descriptions.

Modifier and Type	Method	Description
void	<code>add(int index, E element)</code>	Inserts the specified element at the specified position in this list.
boolean	<code>add(E e)</code>	Appends the specified element to the end of this list.
boolean	<code>addAll(int index, Collection&lt;? extends E&gt; c)</code>	Inserts all of the elements in the specified collection into this list, starting at the specified position.
boolean	<code>addAll(Collection&lt;? extends E&gt; c)</code>	Appends all of the elements in the specified collection to the end of this list.
void	<code>clear()</code>	Removes all of the elements from this list.
Object	<code>clone()</code>	Returns a shallow copy of this ArrayList instance.

# ArrayList Questions

Find the relevant information from Java Docs

- 1: What is the return type of the size() method?
- 2: What does the method size() return?
- 3: How many different ways can the constructor be called?
- 4: What does the subList() method do?

# ArrayList Questions

Find the relevant information from Java Docs

1: What is the return type of the size() method?

int

2: What does the method size() return?

Returns the number of elements in this list.

3: How many different ways can the constructor be called?

3

4: What does the subList() method do?

Returns a view of the portion of this list between the specified fromIndex, inclusive, and toIndex, exclusive



# ArrayList Questions

Find the relevant information from Java Docs

1: What is the name of the library in which ArrayList is contained?

Java.Util

2: What is the initial size of an ArrayList when constructed?

The default size of an ArrayList is 10 until it has an item has been added to it.

3: How would we create an ArrayList with an initial size of 20?

```
private ArrayList<Team> teams = new ArrayList<>(20);
```

4: ArrayList has two versions of the add() method, what is the difference between them?

1 adds an element to the end of the list and returns true if successful, the other inserts an element at a given index and returns void.

# Finally

1. Have a look at the Java Documentation.
2. Search for a few known classes, such as String, Int etc.
3. See if there are any methods you recognise, read what they do and how you can use them in your code.

**It is worth remembering, many of the things you wish to code in your application, probably already exist in a Java library!**