

## Developing Software (SCQF level 8)\_ J6V1 48

**Before we start** - Let's discuss fair use of AI software

**Tip!**– I used AI to generate some of your the guidelines.

I have no problem with you using a similar tool as a "research technique"

**HOWEVER–** You must then extrapolate and combine this with other research so that your answer reflects your own understanding and is refactored into your own wording.

Obviously, It can be an extremely useful tool for grasping and enhancing your understanding of theory (and will be a valid tool to utilise in the workplace – It would therefore be ridiculous to ignore its potential) .

Similarly, it can be used to perhaps give you an example of a code snippet if you are struggling with a particularly tricky concept but should not be used to generate entire passages of your code.

ie: you might ask it to show me an example of how to establish a JDBC connection to a MySQL database. This will give you a guideline example from which you can interpret and adapt – to make your own connection to your own database .

**So, "DO NOT" try to pass off AI generated text or code samples as your own work** this

will not reflect your own understanding, is actually quite easy to detect and could result in you losing your apprenticeship.

Here is what an alternative AI software thought of my answer –

**Yes, the answer provided appears to have been generated by AI software!**

YOU MAY FREELY QUOTE FROM AI GENERATED ANSWERS and discuss the relevance of the information you have researched

**YOU MUST, HOWEVER, REFERENCE THE AI SITE AND DATE OF USE.**

## **Performance Requirements**

### **1. Selecting and applying software development paradigms relevant to the end user context.**

So, quite a big question – tricky! Lol – Here is how I would tackle it:

#### **Guidelines first think what is a software paradigm?**

A software development paradigm is a methodology used to structure and define how we approach software development. Some of the more common paradigms are listed below. **(fuller list at bottom)**

- Procedural
- Event Driven
- Object-Oriented
- Functional

I would research each one to a point that I feel I have a good grasp of each concept.

**Then think how are these relevant to the end user?**

Are we programming a washing machine with a tiny amount of memory (procedural), building an interactive web-site with lots of buttons and drop down lists (event-driven), describing a complex business logic (object-oriented) or streaming & filtering huge amounts of data (functional).

## Now think what are end user contexts?

End user contexts are the scenarios and environments in which the end users interact with the given software system. ( goals, tasks, preferences, devices, and environment)

**So your task is to write a decent paragraph (approx 150-200 words) explaining some of the available paradigms and why we might choose one over the other given different user contexts. You should include a captioned screenshot of one of the web-sites you have used for your research. The caption should explain the relevance of the screenshot to your answer and additionally reference it with the web-site address and date of research.**

example:



**1.1. Software Development Paradigms and Processes**

Videos | Time: 00:00:00 | Download: Large, Large (CC) | Small | Streaming: Streaming (CC) | Slides (PDF)

**Paradigms** consist of "a set of assumptions, concepts, values, and practices that constitutes a way of viewing reality for the community that shares them, especially in an intellectual discipline." When applied to software development, paradigms guide how developers view a problem and organize its solution. Think of it as a language for describing a problem and designing and implementing a software solution. The term *paradigm shift* denotes "a fundamental change in [the] approach or underlying assumptions." Applied to software development, it means an abrupt change in the language between parts of the development process. Computer scientists have used various techniques to help manage the increasing complexity of software systems. They base each technique on a different paradigm, and each advancement caused or resulted in a paradigm shift.

To develop large, complex software systems, developers break the overall development process into smaller, more manageable steps or phases. Jurison<sup>1</sup> notes that "The choice of the software development process has a significant influence on the project's success. The appropriate process can lead to faster completion, reduced cost, improved quality, and lower risk. The wrong process can lead to duplicated work efforts and schedule slips, and create continual management problems."

**The Software Development Process**

Many software development processes have been proposed over time, each defining a specific set of phases or steps. There are typically some variations between the steps or how they are named, but three are common to most processes:

**Analysis**

Developers focus on understanding and modeling the problem during analysis. They form the initial model by abstracting its essential aspects or features and expressing them in a notation that all stakeholders, customers, domain experts, and implementors understand. Their primary goal is to create a model, independent of any programming language or system, that describes the real-world actors and their relationships. The object-oriented paradigm calls this phase *object-oriented analysis (OOA)*, and its end product is a set of connected classes describing the original problem. We'll present our models using a notation specifically created for the task: the *Unified Modeling Language*.

**Design**

The design phase bridges the gap between analysis and implementation by forming a solution architecture. The architecture reflects the domain knowledge discovered and recorded during the analysis phase and defines the framework upon which developers construct the final solution. Programs often require data structures and associated operations that don't exist in the original problem, and developers typically add them during the design phase. They also add details overlooked during analysis and remove irrelevant details during the design phase. The object-oriented paradigm calls this phase *object-oriented design (OOD)*, and, significantly, its end product is the same, albeit refined, set of classes produced by OOA. Developers often partition these classes into sub-models focused on specific, related tasks. Some common groupings are:

- User interface
- Data management
- Task management
- Communications management

Back | Chapter TOC | Next

Caption - Above you can see an article on "Software Development Paradigms and Processes"

<https://icarus.cs.weber.edu/~dab/cs1410/textbook/1.Basics/models.html>

as at 03/07/24

I have given you a list of common paradigms below, hopefully this helps.

### **List of common Paradigms:**

#### Imperative Paradigm

A sequence of instructions that change the state of the system.

C Language

#### Procedural Paradigm

A subset of the imperative paradigm, focus is on capturing the steps of a procedure into functions or subroutines that are subject to procedure calls. procedural programming focuses on the concept of procedure calls.

Pascal, Fortran

#### Event-Driven Paradigm

This paradigm is centred around the concept of events and event handlers. It is commonly used in graphical user interfaces.

JavaScript, Visual Basic

#### Object-Oriented Paradigm

This paradigm is based on the concept of “objects,” which are instances of classes that encapsulate data and behaviour. It promotes code reuse and modularity.

Java, C#, Java, Python

#### Functional Paradigm

In this paradigm, Functional programming is a way of writing software where you build programs using pure functions. Functions always produce the same output for the same input and don't change any outside data, making your code easier to understand and maintain

Haskell and Lisp

## 2. Developing effective user interfaces for the platform being developed.

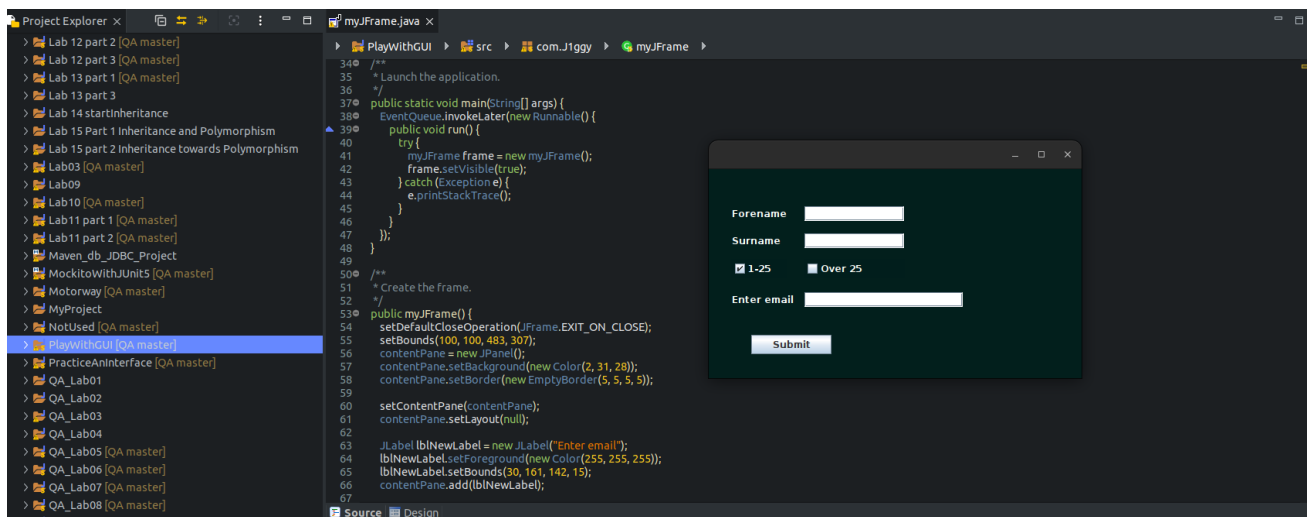
**How to answer** - For our paragraph we could discuss how there are many ways to generate a graphic user interfaces for a Java application. (examples listed below).

Think what does the customer need and how should I implement that?

We should discuss considerations such as usability, intuitiveness, responsiveness(different shaped/sized devices), Accessibility(catering for disability), robustness, research of end user needs and preferences, aesthetics, cross compatibility (different OS's), wire-framing, prototyping and modelling, Documentation and help systems, user and engineer support, performance and security.

We could use “Play with GUI” that we did in class where we used WindowBuilder to generate a simple GUI. This could also provide our screenshots.

ie:



Caption - Above you can see that we have created a Java Application with a simple GUI implemented with WindowBuilder in Eclipse. WindowBuilder is a plugin that supports both Swing and SWT Libraries.

### **List of some suggested methods/Libraries that we could provide a GUI:**

JavaScript frameworks are collections of pre-written JavaScript code that provide developers with a structured and efficient way to build web applications. They offer reusable components and tools to streamline development, enforce best practices, and enhance productivity. (examples incl React, Angular, Vue.js etc)

JSP & Servlets: Servlets are Java classes that extend the capabilities of servers that host applications accessed via an HTML request-response model. JSP is a technology that allows developers to create dynamically generated web pages based on HTML, XML, or other document types. JSP pages are essentially HTML pages with embedded Java code

Swing: A part of Java Foundation Classes (JFC), Swing is a widely used GUI toolkit that provides a rich set of widgets and packages to create sophisticated GUI components. It's platform-independent and lightweight.

JavaFX: This is the modern way to create rich internet applications with a lightweight user interface API. JavaFX provides a more modern and flexible approach compared to Swing and includes features like property binding and CSS styling.

Abstract Window Toolkit (AWT): The original Java GUI toolkit, AWT is platform-dependent and provides basic GUI components. It's less commonly used today but still relevant for simpler applications.

SWT (Standard Widget Toolkit): Developed by IBM, SWT uses native OS components for rendering, which can result in a more native look and feel. It's commonly used with the Eclipse IDE2.

WindowBuilder: An Eclipse plugin that allows you to design GUI applications visually. It supports both Swing and SWT, making it easier to create and manage GUI components.

NetBeans GUI Builder (formerly known as Project Matisse): A powerful tool integrated into the NetBeans IDE that allows you to design GUIs by dragging and dropping components.

JFormDesigner: A commercial GUI designer for Java Swing, it provides a WYSIWYG (What You See Is What You Get) editor and supports various layout managers.

Scene Builder: A visual layout tool for JavaFX applications. It allows you to design user interfaces without writing any code and integrates seamlessly with JavaFX5.

### **3. Writing good quality code (logic) with sound syntax.**

#### **How to answer**

Firstly the importance of well commented code when working in a team environment

Making full use of the IDE and the available automatic tools for generating code – (Intellisense (colour coding, auto completion, hover-over, parameter info, error detection) All help the coder write better code.

Eclipse has “Source” and “Refactor” tools built in that allow you to auto generate Constructors and Getters & Setters, rename elements and auto update the references to that element.

IDE Debugging tools.

Making use of tools like Maven or Gradle to update the project dependencies

Using repositories like Git Hub that can be integrated with the IDE and manage team sharing and version control

We could discuss OOP principles the use of Java Design Patterns reusable tried and tested patterns for solving common issues.

We could discuss TDD Test driven development where we write the tests first and then implement the code to verify the test. (Still to cover but you could research the basic principles)

```

> UserInterfaceMeals2Go > src > main.webapp > XMLParser >
22 * We will need to implement an Adapter Pattern to interpret all these different formats that the restaurants supply their menus in.
23 * But first let's deal with the XML file supplied by Burgers_R_US
24 */
25
26 //CREATE A CLASS TO PARSE THE BURGERS_R_US SUPPLIED XML FILE
27 public class XMLParser {
28
29     private static final String Burgers_R_US_FILENAME = "/home/watso/eclipse_Workspace_JEEE/Parsers/src/com/J1ggy/Menu.xml"; //THIS LINE JUST SETS THE XML FILENAME TO A VA
30     ArrayList<Item> items = new ArrayList<>(); //Creates an ArrayList to hold each menu item
31     // Instantiate the Factory
32     //DocumentBuilderFactory - Defines a factory API that enables applications to obtain a parser that produces DOM object trees from XML documents.
33     DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); //Defines a factory API that enables applications to obtain a parser that produces DOM object trees fr
34     /* ".newInstance()" - Obtains a new instance of a DocumentBuilderFactory. This method uses the JAXP Lookup Mechanism to determine the DocumentBuilderFactory implementatio
35     Once an application has obtained a reference to a DocumentBuilderFactory, it can use the factory to configure and obtain parser instances. */
36
37     try { //Embed in Try Catch block as we may fail to read the XML file in
38
39         // optional, but recommended
40         // process XML securely, avoid attacks like XML External Entities (XXE)
41         factory.setFeature(XMLConstants.FEATURE_SECURE_PROCESSING, true);
42         /*
43         IMPORTANT! - An XML eXternal Entity injection (XXE), which is now part of the OWASP Top 10 via the point A4,
44         is attack against applications that parse XML input. This issue is referenced in the ID 611 in the Common Weakness Enumeration referential.
45         An XXE attack occurs when untrusted XML input with a reference to an external entity is processed by a weakly configured XML parser,
46         and this attack could be used to stage multiple incidents, including:
47             1 - A denial of service attack on the system
48             2 - A Server Side Request Forgery (SSRF) attack
49             3 - The ability to scan ports from the machine where the parser is located
50             4 - Other system impacts.
51         */
52
53         // parse XML file
54         // The Javax.xml.parsers.DocumentBuilderFactory.newInstance() method creates a new instance of a DocumentBuilder using the currently configured parameters.
55

```

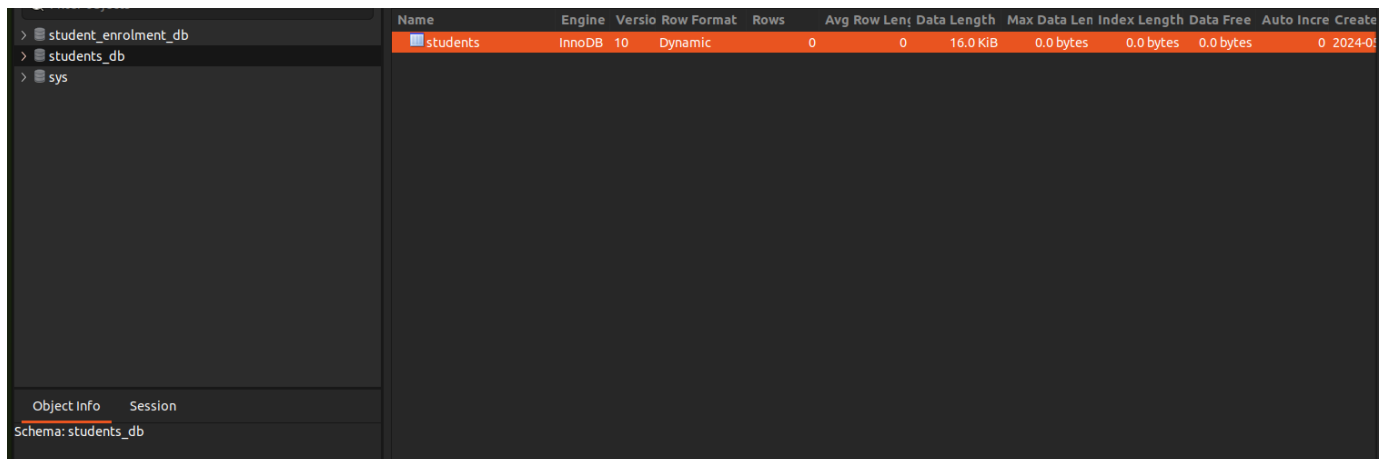
Caption – Above you can see that my code is thoroughly commented. This is to ensure that any other team member who has to pick up the project will be quickly able to understand my implementation. The picture also shows that the IDE's Intellisense has colour coded each line of code appropriately. **(any well commented code snippet will do for screenshot)**



## 4. Linking code to databases and data sets to provide access to data stores.

### How to answer

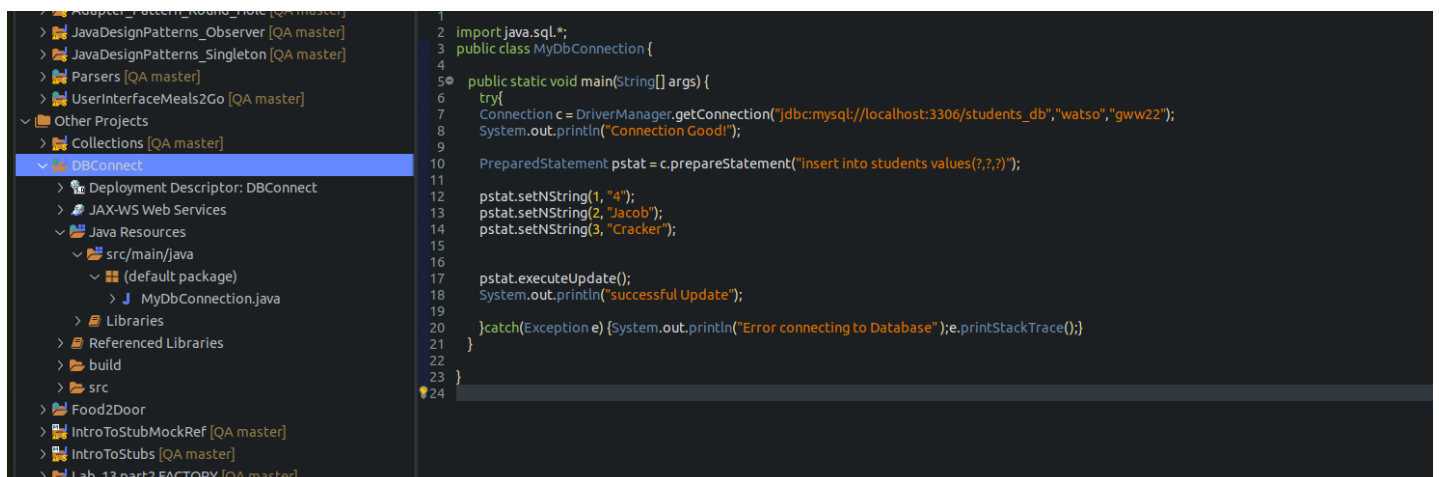
Add two screenshots from the DBConnect we covered in class



The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' pane lists 'student\_enrolment\_db', 'students\_db', and 'sys'. The 'students\_db' schema is selected. Below it, the 'Object Info' tab shows 'Schema: students\_db'. The main area displays a table named 'students' with the following columns: Name, Engine, Version, Row Format, Rows, Avg Row Len, Data Length, Max Data Len, Index Length, Data Free, Auto Increment, and Create Time. The table is empty, with 0 rows.

Name	Engine	Version	Row Format	Rows	Avg Row Len	Data Length	Max Data Len	Index Length	Data Free	Auto Increment	Create Time
students	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0	2024-01-01 12:00:00

Caption - Here you can see I have created a MySQL database called “students\_db” with a students table. Screenshot from MySQLWorkbench GUI.



The screenshot shows an IDE with a project explorer on the left and a code editor on the right. The project explorer shows a project named 'DBConnect' with a file named 'MyDbConnection.java'. The code editor shows the following Java code:

```
1 import java.sql.*;
2 public class MyDbConnection {
3
4
5     public static void main(String[] args) {
6         try{
7             Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3306/students_db","watso","gww22");
8             System.out.println("Connection Good!");
9
10            PreparedStatement pstmt = c.prepareStatement("insert into students values(?,?,?)");
11
12            pstmt.setString(1, "4");
13            pstmt.setString(2, "Jacob");
14            pstmt.setString(3, "Cracker");
15
16            pstmt.executeUpdate();
17            System.out.println("successful Update");
18        } catch (Exception e) {System.out.println("Error connecting to Database");e.printStackTrace();}
19    }
20 }
21
22
23
24
```

Caption - Here you can see a screenshot of the “myDbConnection” class which makes a connection to the database and executes a prepared statement to add a student to the table.

For the paragraph discuss what each line in the “myDbConnection class is doing.

**The class Listing – Explain each line of code**

```
import java.sql.*;

public class MyDbConnection {

    public static void main(String[] args) {
        try{
            Connection c =
DriverManager.getConnection("jdbc:mysql://localhost:3306/students_db", "wats
o", "gww22");

            System.out.println("Connection Good!");

            PreparedStatement pstat = c.prepareStatement("insert into
students values(?,?,?)");

            pstat.setString(1, "5");
            pstat.setString(2, "Heinz");
            pstat.setString(3, "Tomato");

            pstat.executeUpdate();
            System.out.println("successful Update");

        }catch(Exception e) {System.out.println("Error connecting to
Database" );e.printStackTrace();}
    }
}
```

#	studentID	firstName	lastName
1	1	Mohammed	Ali
2	2	Joe	Fraser
3	4	Jacob	Cracker
4	5	Heinz	Tomato
*	NULL	NULL	NULL

Caption – Here you can see student “Heinz Tomato” has been added to the database

## 5. Applying agreed standards and tools to achieve well-engineered software including code. commenting, naming and layout.

### How to answer

Add a paragraph discussing the importance of consistency of code writing within a software development team. Discuss the importance of commenting and documenting your work. Write an example of a Java Class for a student with **ID, first name, last name, age, Tel, e-mail, Address, Post code, courses** and **achievements**. Think about appropriate data types for each field (**You should remind yourself how to implement ArrayLists**) Implement a Constructor with parameters and “Getters & Setters”. Write a program that creates and instantiates one instance of your class and prints out the details using an overridden toString() method (**remember to call super()**). Comment each line of your code. Discuss your choices and the naming conventions you have used. Follow Oracles naming conventions throughout this exercise. (**Link below**)

Screenshot and Caption your Student Class

ORACLE

Products Industries Resources Customers Partners Developers Company

Q

🇺🇸

👤

View Accounts

📄

Contact Sales

9 - Naming Conventions

Naming conventions make programs more understandable by making them easier to read. They can also give information about the function of the identifier-for example, whether it's a constant, package, or class-which can be helpful in understanding the code.

Identifier Type	Rules for Naming	Examples
Packages	<p>The prefix of a unique package name is always written in all-lowercase ASCII letters and should be one of the top-level domain names, currently com, edu, gov, mil, net, org, or one of the English two-letter codes identifying countries as specified in ISO Standard 3166, 1981.</p> <p>Subsequent components of the package name vary according to an organization's own internal naming conventions. Such conventions might specify that certain directory name components be division, department, project, machine, or login names.</p>	<pre>com.sun.eng com.apple.quicktime.v2 edu.cmu.cs.bovik.cheese</pre>
Classes	Class names should be nouns, in mixed case with the first letter of each internal word capitalized. Try to keep your class names simple and descriptive. Use whole words-avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as URL or HTML).	<pre>class Raster; class ImageSprite;</pre>
Interfaces	Interface names should be capitalized like class names.	<pre>interface RasterDelegate; interface Storing;</pre>
Methods	Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.	<pre>run(); runFast(); getBackground();</pre>
Variables	<p>Except for variables, all instance, class, and class constants are in mixed case with a lowercase first letter. Internal words start with capital letters. Variable names should not start with underscore _ or dollar sign \$ characters, even though both are allowed.</p> <p>Variable names should be short yet meaningful. The choice of a variable name should be mnemonic- that is, designed to indicate to the casual observer the intent of its use. One-character variable names should be avoided except for temporary "throwaway" variables. Common names for temporary variables are <code>i</code>, <code>j</code>, <code>k</code>, <code>m</code>, and <code>n</code> for integers; <code>c</code>, <code>d</code>, and <code>e</code> for characters.</p>	<div> <div>Copy</div> <pre>int i; char c; float mywidth;</pre> </div>
Constants	The names of variables declared class constants and of ANSI constants should be all uppercase with words separated by underscores ("_"). (ANSI constants should be avoided, for ease of debugging.)	<pre>static final int MIN_WIDTH = 4; static final int MAX_WIDTH = 999; static final int GET_THE_CPU = 1;</pre>

Caption – Here you can see Oracles Code Conventions for naming Java Elements

<https://www.oracle.com/java/technologies/javase/codeconventions-namingconventions.html> as at 03/07/24

## 6. Refactoring software to improve its structure, legibility, efficiency and reusability.

An example of refining our code might be replacing multiple “If else” statements with a “switch” statement.

Take your student class from the previous exercise and add a new **String field** called **“funding”**. (Remember the getter & setter for funding)

We will use this new field and the **“age”** and **“achievements”** fields for an exercise in refactoring code.

## Exercise:

If a student instance is over 25 and they have **“no”** achievements then they should be funded by a **“backToWrkGrant”** otherwise they should be funded by a **“studentLoan”**.

Using the following keywords:

- **if** - to specify a block of code to be executed, if a specified condition is true
- **else** - to specify a block of code to be executed, if the same condition is false
- **else if** - to specify a new condition to test, if the first condition is false

Re-implement your program to include a student instance (**Andy Mann who is 26 and has no achievements**) and a student instance (**Abe Young who is 24 and has achieved “Html\_CSS\_JScript\_Module\_1”**)

Your program should use an adjusted **“toString()”** method to print out each students details.

Screenshot and caption your code showing both the code and your Console result.

The challenge is to now refactor your code so that you no longer make use of **“if else”** constructs” but instead use a **“switch statement”** to make all the necessary checks.

Re-screenshot and caption your new refactored code showing both the code and your Console result.

Discuss your refactoring

## **7. Reviewing own software development activities to find and eliminate problems and identify productivity improvements.**

### **How to answer**

Think of an example from your course work (or workplace) so far where you have either hit an issue, had to use the debugger, missed a library ... Anything that has stopped you in your tracks and that you have had to resolve. Write a paragraph discussing what the issue was, how you resolved it and what you might do differently.

Discuss your use of comments when you review some work which you haven't looked at for a few weeks ...Are your comments strong enough?

Have you researched the technologies such as GitHub or Maven that help you configure your project set up or dependencies, store, share and version control your code? Discuss your experiences.

Create a GitHub Account and practice tracking your work

### **You have access to QA online content to enhance your learning.**

<https://cloudacademy.com/login>

**[Java Servlet Technology in Java EE 7 Lesson \(qa.com\)](#)**    QA content

Does your employer help review your work? Discuss that experience.

Do you attend "Scrum" meetings or review meetings at your work?

### **Optional**

Try creating a Maven project – using your research resources and see if you can add the dependencies you need from the Maven Repository

<https://mvnrepository.com>

To do this choose an area of research such as Servlets and JSP.

There are lots of short courses freely available on YouTube that can provide “alternative takes” or just a fun challenge if you have the spare time. I thought this one was quite good - <https://www.youtube.com/watch?v=b42CJ0r-1to&list=PLE0F6C1917A427E96>

How good an experience do you get from pushing yourself to research & self learn?

The screenshot shows the Maven Repository search results for the term "Servlet". The interface includes a sidebar with filters for Repository, Group, and Category. The main content area displays a list of search results, with the top four results being:

- 1. Java Servlet API**  
javax.servlet » javax.servlet-api  
Java Servlet is the foundation web specification in the Java Enterprise Platform. Developer workflow.  
Last Release on Apr 20, 2018  
**Relocated** → jakarta.servlet » jakarta.servlet-api
- 2. JavaServlet(TM) Specification**  
javax.servlet » servlet-api  
Java Servlet is the foundation web specification in the Java Enterprise Platform. Developer workflow.  
Last Release on Jul 17, 2006  
**Relocated** → javax.servlet » javax.servlet-api
- 3. Jakarta Servlet**  
jakarta.servlet » jakarta.servlet-api  
Jakarta Servlet  
Last Release on Jun 11, 2024
- 4. Jetty :: Servlet Handling**  
org.eclipse.jetty » jetty-servlet  
Jetty support for the Servlet APIs and implementation of the behaviors required by the Serv

Caption - Above you can see the Maven Repository where I am searching for a Java Servlet API. The Maven Repository is “a standard way to build the projects, a clear definition of what the project consisted of, an easy way to publish project information, and a way to share JARs across several projects”.Quoted from <https://maven.apache.org/what-is-maven.html>

## **8. Collaborating with others in work reviews to support improvements to the software development processes adopted.**

Group project – We will split the class into groups (of approx 4).

You can decide amongst yourselves how you wish to communicate – WhatsApp, Teams, email etc. Each group should elect a Project Manager who will set up the communications and monitor the the input. You should create a GitHub group repository shared for each member.

The Project itself will be to create a database driven course enrolment system where students and tutors can create an account.

We can create a Course(courseID ,name, price and departmentID)

We can enrol a Student(studentID, firstName, lastName, enrolmentDate, Email, houseNo, postcode, pin)

Items can be added to a cart(cartID, StudentID, courseID, orderDate)

Relationships

A student can have many cartItems but a cartItem is associated with one student.

...a course can be associated with many cartItems but a cartItem can only be one course

**(Nb: It should not actually be published on line but should run on a Tomcat Web-Server using Servlets & JSPs)**

The business logic will be to display a Login or redirect to a Register page.

If Login is successful start a Session and display the courses.

Else Register – create a student notify them of their pin and redirect to Login



The courses should display as a table and allow the students to filter & sort the table. They should also be able to select a course and add/remove it to/from their cart.

Build a schedule and agree on meeting times.

Decide on an approach – Waterfall, Agile, Scrum etc

Jointly research Servlets and JSPs.

### **Java Servlet Technology in Java EE 7 Lesson (qa.com)**

<https://www.youtube.com/watch?v=b42CJ0r-1to&list=PLE0F6C1917A427E96>

Create an Entity Relationship Diagram for your database.

Model the graphic user interfaces for approval.

Create a set of UML Diagrams to model the system requirements.

Delegate tasks between Team members setting appropriate timescales.

Track the teams progress – keeping records of who has done what and by when. This can be an Excel spreadsheet showing a breakdown of tasks, estimated and actual completion times of each task and the responsible person for each task.

Please report back to your tutor if anyone is not pulling their weight at the earliest opportunity. We can then approach them and try to resolve any issues as quickly as possible so that this is not detrimental to team progress. In extreme circumstances we may remove a team member.

The final submittal of evidence should include all your modelling diagrams, a full code listing, A project tracking record as described and a demonstration video of one of the candidates using all of the features of the software. This can be as simple as a video recoded on a phone. We

can use any suitable free video editor such as <https://www.wevideo.com> to tidy this up. If you require assistance with this step please request through your assessor.

Finally – Research and produce a set of support documentation for your project. Include a bibliography of all research books, websites etc used by the team whilst carrying out your research or to aid in resolving/improving your code.

Nb: FEEL FREE TO INVITE YOUR TUTOR TO YOUR MEETING \_ IF I CAN ATTEND I WILL ENDEAVOUR TO HELP :D

**9. Using third party integration tools (screen scraping data and integrating to other systems) to capture, reformat and display data more conveniently.**

DEFER FOR LATER \_ MATERIAL NOT YET COVERED

**10. Staging and deploying validated code into live enterprise environments.**

DEFER FOR LATER \_ MATERIAL NOT YET COVERED

**11. Debugging code and applying structured techniques to problem solving.**

**How to answer -**

Write a small app of your own choice introduce an issue (perhaps out of index on an array) capture screenshots of the issue using the debugger provided in Eclipse. Fix your code with the use of Error Handling (Try catch) provide screenshots and a paragraph explaining how you used the debugger to resolve the problem

**Suggested research video:**

<https://www.bing.com/videos/riverview/relatedvideo?>

[q=Eclipse+Debugging+tutorial&mid=D96743B5CF259391B48CD96743B5CF259391B48C&FORM=VIRE](https://www.youtube.com/watch?v=9TuKt5G4MVo)

## **12. Describe how to read and write technical software documentation.**

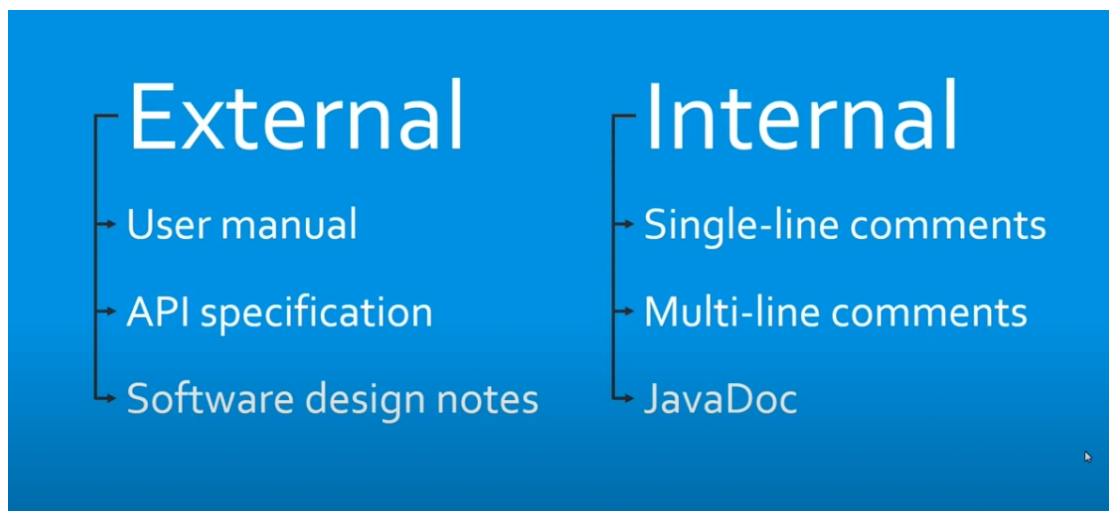
### **How to answer -**

Discuss internal and external documentation

Watch the following video

<https://www.youtube.com/watch?v=9TuKt5G4MVo>

Implement some JavaDocs for one of your code samples – screen shot and describe the difference between Internal and External Documentation and the importance and benefits.



## **13. Describe the SDLC software Development Life Cycle and each of its stages**

### **How to answer -**

Information covered by course slides

See - “Software Developer Level 8 Designing Software Day 1”

## **14. Discuss some of the Range of development tools available and how you have used them.**

### **How to answer -**

**DEFER FOR NOW -**

Over the course of the year will be using a number of software tools and plug-ins including the Eclipse IDE, Junit, WindowBuilder, Selenium Webdriver, JDBC drivers, MySql, Maven, Git-Hub, Servlets, JSP, SpringBoot, Visual Paradigm etc... I think this question is therefore best answered towards the end of the course.

**15. Discuss Industry standard software and web design and accessibility frameworks and guidelines and how to apply them, including those from W3C (world wide web consortium).**

**How to answer - Research** (Sample answer below to be replaced with own version)

**The World Wide Web Consortium (W3C)** plays a crucial role in establishing these standards. W3C's Web Content Accessibility Guidelines (WCAG) provide a set of recommendations for making web content more accessible to people with disabilities. To apply these guidelines effectively, designers and developers should consider factors such as providing alternative text for images, ensuring proper heading structures for screen readers, implementing keyboard navigation, and maintaining colour contrast for readability. Additionally, utilizing responsive design principles ensures websites are accessible across various devices. By following these industry standards and guidelines, developers can create software and websites that are not only visually appealing but also functional for all users, including those with disabilities.

**ONCE YOU HAVE ALL YOUR ANSWERS COMPLETED**

PLEASE NOW CHECK YOUR ANSWERS AGAINST THE MAPPINGS BELOW TO ENSURE THAT YOU HAVE COVERED ALL THE EXPECTED KNOWLEDGE CRITERIA IN GIVING YOUR ANSWERS – I HAVE TRIED TO INDICATE WHERE WE WOULD EXPECT YOU TO HAVE COVERED THESE IN YOUR ANSWERS, HOWEVER, IF YOU FEEL YOU ARE MISSING ANYTHING PLEASE TRY TO RE-VISIT THE QUESTION AND APPEND A COUPLE OF LINES TO COVER YOURSELF>

## Knowledge and understanding

1. How to review software requirements and design specifications.  
- (covered by Q1)
2. How to read and write technical software documentation.  
- (covered by Q12)
3. The software development lifecycle.  
- (covered by Q13)
4. How to operate at all stages of the software development lifecycle.  
- (covered by Designing Software (SCQF level 8)\_J6TY48 module see separate essay)
5. Good practice approaches for the relevant software development paradigm, including object oriented, event driven or procedural.  
- (covered by Q1)
6. How to develop software for web, mobile and fixed platforms.  
- (covered by Q2)
7. The range of development tools available and how to use them.  
- (covered by Q2 & Q14)
8. How to produce software code directly using a command line editor.  
- Covered in class by Lab - (A Guide to Running a Java Application From The Command Line Software Level 8)  
  
Follow Lab guide and paste in captioned screenshot
9. Industry standard software and web design and accessibility frameworks and guidelines and how to apply them, including those from W3C (world wide web consortium).  
- (covered by Q15)
10. How to implement unit testing at each stage of software development.  
- (covered by Providing Software Testing and Assurance\_ J6VA 48 module )
11. How to develop software using industry standard software languages, and development environments and tools.  
- (covered in detail across 3 modules -
  - Designing Software (SCQF level 8)\_J6TY48 module
  - Developing Software (SCQF level 8)\_J6V1 48
  - Providing Software Testing and Assurance\_ J6VA 48 )

12. How to resolve software development problems through online research to find solutions.  
- (covered by Group Project Q8)
13. How to design, build and interface with databases to provide data creation, updating and deletion functions.  
- (covered by Group Project Q4)
14. How teams work effectively to produce software.  
- (covered by Q8 & the Group Project Designing Software (SCQF level 8)\_J6TY48 module)
15. The importance of considering different approaches and tools including cost and efficiency.  
(covered in detail across 3 modules -
  - Designing Software (SCQF level 8)\_J6TY48 module
  - Developing Software (SCQF level 8)\_J6V1 48
  - Providing Software Testing and Assurance\_J6VA 48 )