JAVA WORKSHOP

Day 1 – 30/11

Introductions – SLIIT/FoC

Introductions – Resource Persons

- **Mr. Dilanka Nayanajith**
- Academic Instructor
  Department of Information Technology
  Faculty of Computing
- Demonstrator
  IT2030 – Object Oriented Programming (Y2S1 compulsory)
  100% Java based Module
- BSc. Special (Hons) Degree in Information Technology
  (Specialized in Information Technology) at SLIIT
- Experience:
  Academic Instructor – SLIIT
  Software Developer – Synapse Solutions

Introductions – Resource Persons

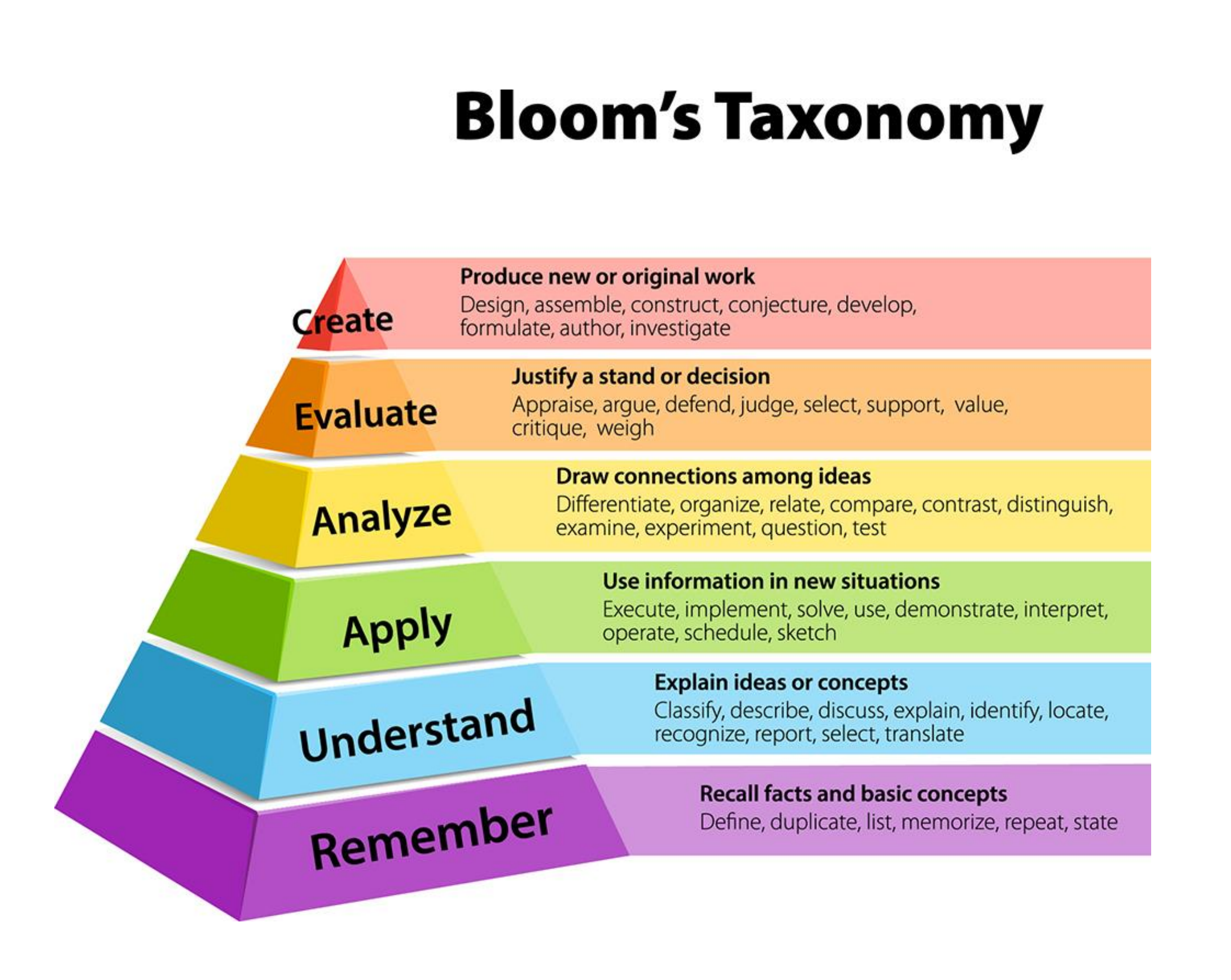
- **Ms. Thilmi Anuththara Kuruppu**
- Lecturer
  Department of Computer Science & Software Engineering
  Faculty of Computing
- Lecturer In Charge
  IT2030 – Object Oriented Programming (Y2S1 compulsory)
  100% Java based Module
- BSc. Special (Hons) Degree in Information Technology
  (Specialized in Software Engineering) at SLIIT (2010-2013)
  First-Class
- Experience:
  Academic Researcher, Lecturer, Instructor – SLIIT
  Associate Software Engineer – Pearson Lanka
  Business Analyst – Total Amber Ltd.

Introductions – Resource Persons

- **Dr. Shyam Reyal**
- Senior Lecturer, Department of Information Technology, Faculty of Computing
- Lecturer In Charge
  COMP2003 – Object Oriented Software Engineering* (Y2S1)
  COMP3003 – Software Engineering Concepts* (Y3S2)
  *100% Java based modules
- BSc. (Hons) Engineering (Specialized in Computer Science & Engineering) at University of Moratuwa (2006-2010) – First Class
- PhD in Computer Science (Specialized in Human Computer Interaction) at University of St Andrews (2012-2019) – SICSA Prize Scholarship
- SCJP, SCMAD, SCWCD, SCBCD, SCEA (2005-2008)
- Virtusa Inc. – Sri Lanka
  University of St Andrews – UK
  NOMAD Inc. – UK
  Google. Inc. – Mountain View, CA, USA
  University of Cambridge, UK

Important Note!

- **This is a crash course!**
  - The target content is taught over at least 2 semesters in a full-time degree program
- We will try our best to achieve the learning outcomes in five days:
  - Explain the theory and concepts in Java Programming (**BT2**)
  - Develop a software solution using Java (**BT3**) for a given problem (**BT4**)
  - Select the best possible Java based technology or platform (**BT5**) for a given problem (**BT4**)


Bloom's Taxonomy
Create
Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate
Evaluate
Justify a stand or decision
Appraise, argue, defend, judge, select, support, value, critique, weigh
Analyze
Draw connections among ideas
Differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test
Apply
Use information in new situations
Execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch
Understand
Explain ideas or concepts
Classify, describe, discuss, explain, identify, locate, recognize, report, select, translate
Remember
Recall facts and basic concepts
Define, duplicate, list, memorize, repeat, state

Daily Agenda

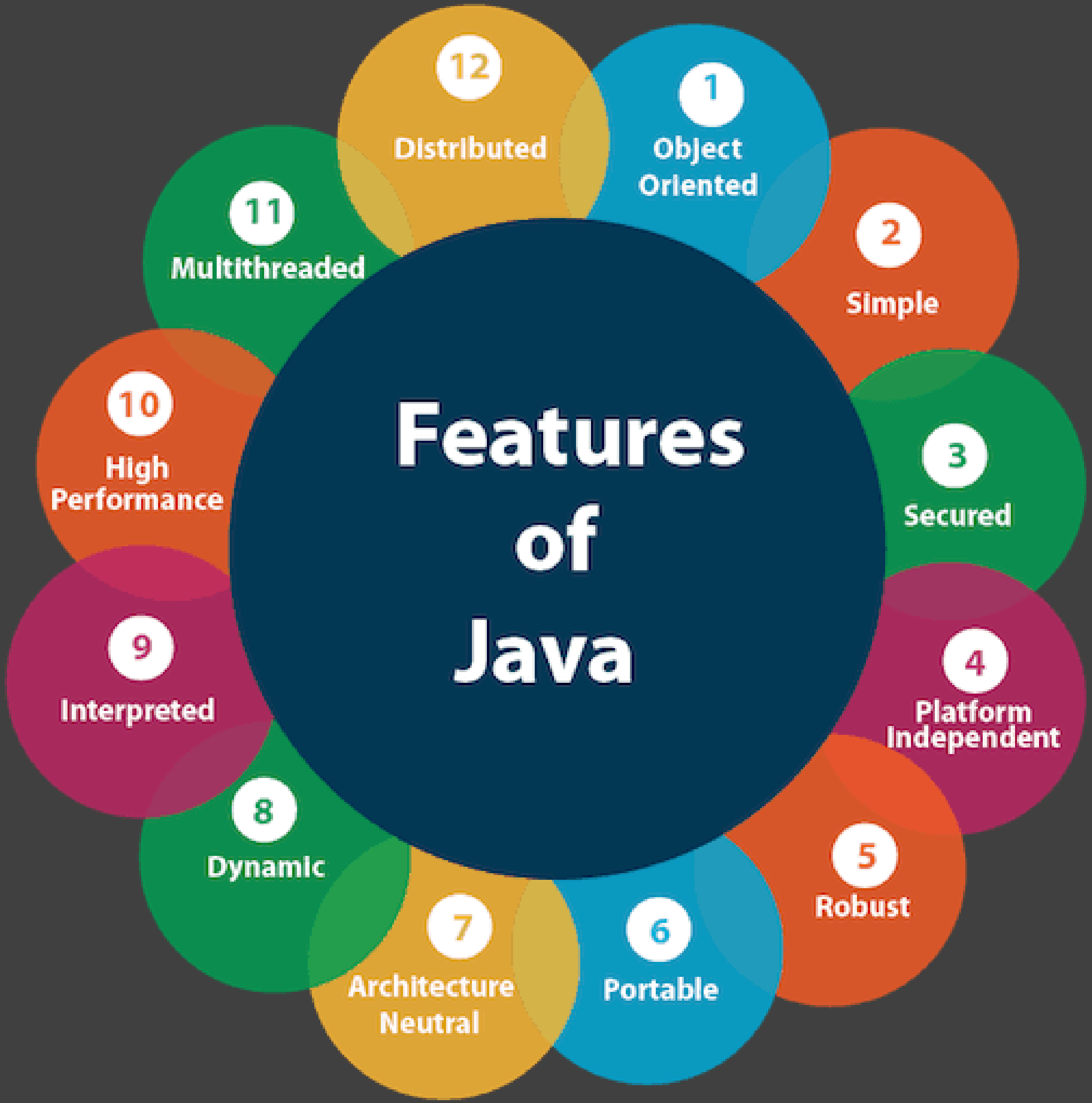
Time Slot	**Session**	**Duration**
9.00am – 10.30am	Session 1	1.5h
10.30am – 11.00am	Tea	½ h
11.00am – 12.30pm	Session 2	1.5h
12.30pm – 1.30pm	Lunch	1h
1.30pm – 3.00pm	Session 3	1.5h
3.00pm – 3.30pm	Tea	½ h
3.30pm – 5.00pm	Session 4	1.5h

Initial Survey

- As a trainer, I must first understand my audience.
- First, **I would like to know each one of you.**

- Next, **please fill this survey** to understand your domain knowledge level https://docs.google.com/forms/d/e/1FAIpQLSfAROCmyazusLUWFS9SftU-4LbXywfp8rlUBhRFNTH_sQBZ7Q/viewform?usp=sf_link
- This will help me understand exactly the level which I should teach at, and cater my examples.

WHY JAVA?


Features of Java
1 Object Oriented
2 Simple
3 Secured
4 Platform Independent
5 Robust
6 Portable
7 Architecture Neutral
8 Dynamic
9 Interpreted
10 High Performance
11 Multithreaded
12 Distributed

History and Evolution

- Created by James Gosling in Sun Microsystems in mid 90s
- Sun Microsystems was acquired by Oracle in 2010
- Java has undergone 17 version changes
- https://en.wikipedia.org/wiki/Java_version_history
- Since 2007, was handed over to the community under the GPL2.0 license

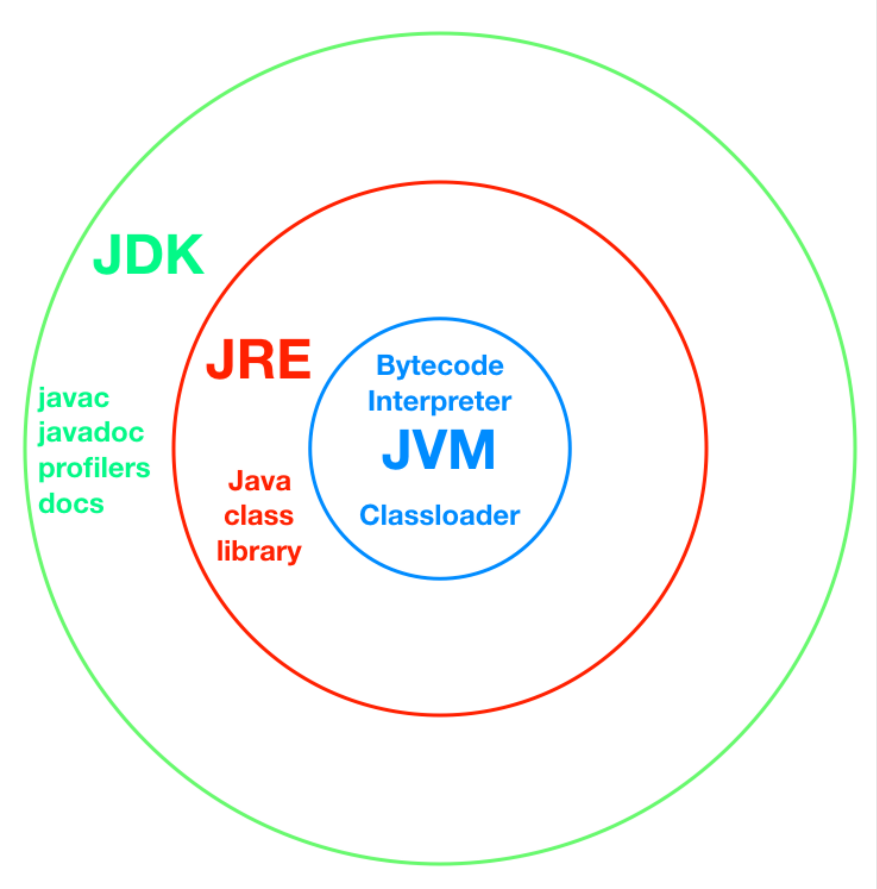
JDK Beta	1995
JDK 1.0	January 1996
JDK 1.1	February 1997
J2SE 1.2	December 1998
J2SE 1.3	May 2000
J2SE 1.4	February 2002
J2SE 5.0	September 2004
Java SE 6	December 2006
Java SE 7	July 2011
Java SE 8 (LTS)	March 2014
Java SE 9	September 2017
Java SE 10	March 2018
Java SE 11 (LTS)	September 2018
Java SE 12	March 2019
Java SE 13	September 2019
Java SE 14	March 2020
Java SE 15	September 2020
Java SE 16	March 2021
Java SE 17 (LTS)	September 2021

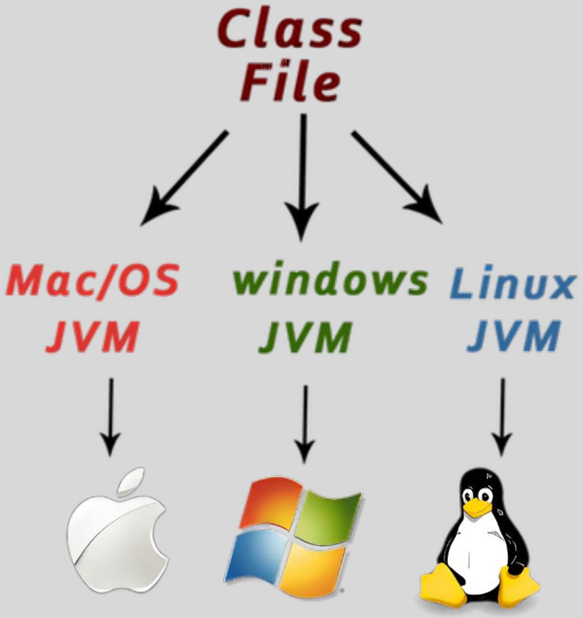
Flavours & Industrial Applications

- The Old Flavours
    - Java-SE
    - Java-EE
    - Java-ME
    - Java-Card
- The New Flavours
    - (some of the above+)
    - Android
    - Android Auto
    - JavaFX

Popularity of Java

- Write-Once-Run-Anywhere (**WORA**)
- Write-Once-Deploy-Anywhere (**WODA**)
- Backward Compatibility

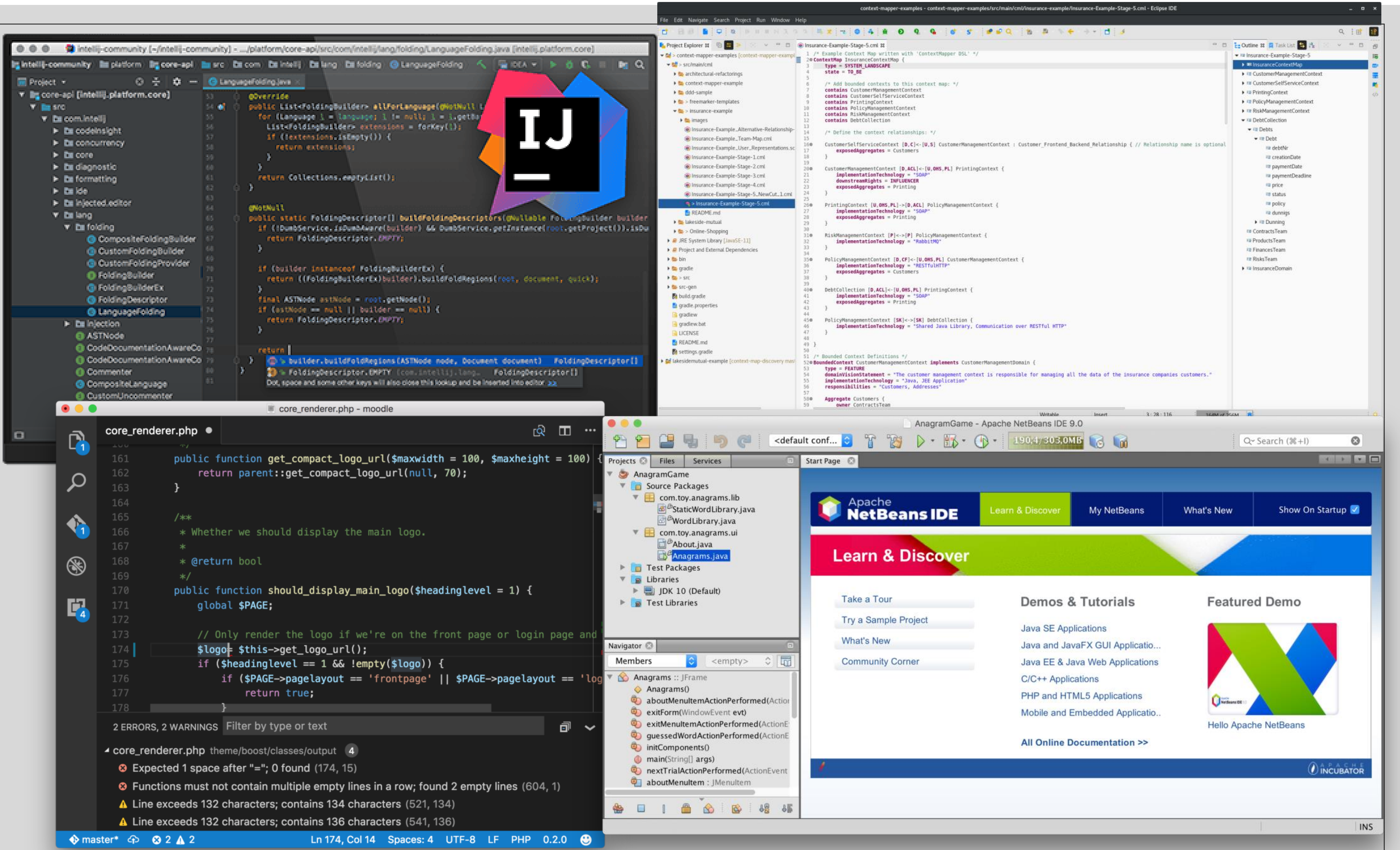




Installing and Configuring Java

- Which version? LTS? Open?
- Be careful about platform and third-party support
  - e.g., Amazon AWS does not support latest JDKs, recommended JDK 11
  - Some IDEs do not work with the latest JDK
- JDK vs JRE
- https://www.oracle.com/java/technologies/downloads/
- https://dev.java/
- https://openjdk.java.net/
- JAVA_HOME and PATH variables

IDEs



```
@Override
public List<FoldingBuilder> allForLanguage(@NotNull L
  for (Language l = language; l != null; l = l.getBa
    List<FoldingBuilder> extensions = forKey(l);
    if (!extensions.isEmpty()) {
      return extensions;
    }
  }
  return Collections.emptyList();
}

@NotNull
public static FoldingDescriptor[] buildFoldingDescriptors(@Nullable Fo   gBuilder builder
  if (!DumbService.isDumbAware(builder) && DumbService.getInstance(root.getProject()).isDu
    return FoldingDescriptor.EMPTY;
  }

  if (builder instanceof FoldingBuilderEx) {
    return ((FoldingBuilderEx)builder).buildFoldRegions(root, document, quick);
  }
  final ASTNode astNode = root.getNode();
  if (astNode == null || builder == null) {
    return FoldingDescriptor.EMPTY;
  }

  return |
  }  builder.buildFoldRegions(ASTNode node, Document document)   FoldingDescriptor[]
     FoldingDescriptor.EMPTY (com.intellij.lang...   FoldingDescriptor[]
     Dot, space and some other keys will also close this lookup and be inserted into editor >>
}
```

```
/* Example Context Map written with 'ContextMapper DSL' */
ContextMap InsuranceContextMap {
    type = SYSTEM_LANDSCAPE
    state = TO_BE

    /* Add bounded contexts to this context map: */
    contains CustomerManagementContext
    contains CustomerSelfServiceContext
    contains PrintingContext
    contains PolicyManagementContext
    contains RiskManagementContext
    contains DebtCollection

    /* Define the context relationships: */

    CustomerSelfServiceContext [D,C]<-[U,S] CustomerManagementContext : Customer_Frontend_Backend_Relationship { // Relationship name is optional
        exposedAggregates = Customers
    }

    CustomerManagementContext [D,ACL]<-[U,OHS,PL] PrintingContext {
        implementationTechnology = "SOAP"
        downstreamRights = INFLUENCER
        exposedAggregates = Printing
    }

    PrintingContext [U,OHS,PL]->[D,ACL] PolicyManagementContext {
        implementationTechnology = "SOAP"
        exposedAggregates = Printing
    }

    RiskManagementContext [P]<->[P] PolicyManagementContext {
        implementationTechnology = "RabbitMQ"
    }

    PolicyManagementContext [D,CF]<-[U,OHS,PL] CustomerManagementContext {
        implementationTechnology = "RESTfulHTTP"
        exposedAggregates = Customers
    }

    DebtCollection [D,ACL]<-[U,OHS,PL] PrintingContext {
        implementationTechnology = "SOAP"
        exposedAggregates = Printing
    }

    PolicyManagementContext [SK]<->[SK] DebtCollection {
        implementationTechnology = "Shared Java Library, Communication over RESTful HTTP"
    }

}

/* Bounded Context Definitions */
BoundedContext CustomerManagementContext implements CustomerManagementDomain {
    type = FEATURE
    domainVisionStatement = "The customer management context is responsible for managing all the data of the insurance companies customers."
    implementationTechnology = "Java, JEE Application"
    responsibilities = "Customers, Addresses"

    Aggregate Customers {
        owner ContractsTeam
```

```
    public function get_compact_logo_url($maxwidth = 100, $maxheight = 100) {
        return parent::get_compact_logo_url(null, 70);
    }


    /**
     * Whether we should display the main logo.
     *
     * @return bool
     */
    public function should_display_main_logo($headinglevel = 1) {
        global $PAGE;

        // Only render the logo if we're on the front page or login page and
        $logo = $this->get_logo_url();
        if ($headinglevel == 1 && !empty($logo)) {
            if ($PAGE->pagelayout == 'frontpage' || $PAGE->pagelayout == 'log
                return true;
            }
```

```
2 ERRORS, 2 WARNINGS   Filter by type or text
core_renderer.php theme/boost/classes/output 4
  ⊗ Expected 1 space after "="; 0 found (174, 15)
  ⊗ Functions must not contain multiple empty lines in a row; found 2 empty lines (604, 1)
  ⚠ Line exceeds 132 characters; contains 134 characters (521, 134)
  ⚠ Line exceeds 132 characters; contains 136 characters (541, 136)
master*   ⊗ 2 ⚠ 2   Ln 174, Col 14   Spaces: 4   UTF-8   LF   PHP   0.2.0
```

Configuring and Installing NetBeans

- We will use Apache NetBeans 12.5, compatible with JDK 16
- https://netbeans.apache.org/download/index.html
- Once installed, you need enable plugins for:
  - Java SE with Ant
  - Java EE Web
- Install a suitable Application Server for Java Web (covered later)

How to write a basic Java program

- Minimum requirements for JavaSE platform
  - At least ONE class
  - At least ONE main method

- Entry points are different for different platforms
  - JavaME – startApp() method
  - JavaEE – JSP/Servlets – doGet() or doPost() methods
  - JavaFX – start() method
  - Android – onCreate() method

- Use the correct method for correct platform!

Operators

- Assignment (=)
- Arithmetic (+ - / * %)
- Unary (++ --)
- Relational (> < >= <= == !=)
- Logical (& | ^)
- Bitwise (~ ! >> << >>>)
- Compound assignment (+= -= /= *= %=)
- Conditional/ternary: (condition)?(value_if_true):(value_if_false);
- Type comparison operators (instanceof)

Variables

- Can be primitive data types or objects
- Primitive Data Types in Java

byte	8 bit	Integral numbers	-128	127
short	16 bit		-32768	32767
int	32 bit		-2^{31}	$2^{31} - 1$
long	64 bit		-2^{63}	$2^{63} -1$
float	32 bit	Floating Point Numbers	infinity	
double	64 bit			
char	16 bit	Characters	‘A’, ‘B’, ‘C’, etc.	
boolean	1 bit	true/false only	false	true

- Object types can be String, or a type of Class

Classes & Objects

- Analyze the following scenario and prepare classes and objects.
- A student has a name, student id number, and age. Teachers have a name and employee number. There are two types of students – Undergraduate and Postgraduates. Undergraduates have a year (Y1-Y4), and postgraduates have a previous degree. John and Jane are two students. John is an undergraduate. Jane is a postgraduate. Smith is a teacher. John's student id is 1, Jane's student id is 2. John is 21 years old and is in the second year. Jane is 23 years old. Smith has an employee number of 1000. Jane studied "BSc. in IT" for her previous degree.
- Analyze this scenario and draw (a) UML class diagram (b) UML object diagram
- Write code in Java

File Handling

- Prerequisite:
  - Getting the console input from the user – use the Scanner class with System.in stream
- Basic file handling operations:
  - Checking if a file exists
  - Creating a new file
  - Reading from a file
  - Writing to a file
  - Listing Files
  - (!!) Deleting Files
  - (!!) Moving Files

Practical 1

- Build a console-based application using Java, to:
- Basic Requirements:
  - Check if a file exists with a given name
  - Create a new file with a given name
  - Read from a file
  - Write to a file
  - List the files/sub-folders in a given folder
- Extended Requirements:
  - Count the number of characters in a file
  - Count the number of lines in a file
  - Count all lines in all files in a folder

SLIIT
Discover Your Future

MDTU JAVA Workshop
Worksheet 01

Introduction to NetBeans

To complete the work sheets, you will use the Apache NetBeans (12.5) as IDE (Integrated Development Environment) and JDK (Java Development Kit) Version 16 as the development Environment.

Virtual machines which you receive already installed above software's

1. Open NetBeans IDE

Start Menu -> NetBeans

Apache NetBeans IDE

(IF there is a desktop shortcut you can use that)

2. Create New Java Project

File –> New Project

What did we learn today?

- Introduction to Java
- Introduction to NetBeans
- Java Programming Essentials
- Object Oriented Programming Basics
- File Handling
- Building console applications

JAVA WORKSHOP

Day 2 - 01/12

What Did We Learn Yesterday

- Recap of previous day’s material
- Clarify doubts from previous day
- Check previous days homework

Conditionals and Control Statements

- Conditionals
  - if/else
  - switch/case

- Control statements
  - for/while/do-while
  - break/continue

Strings

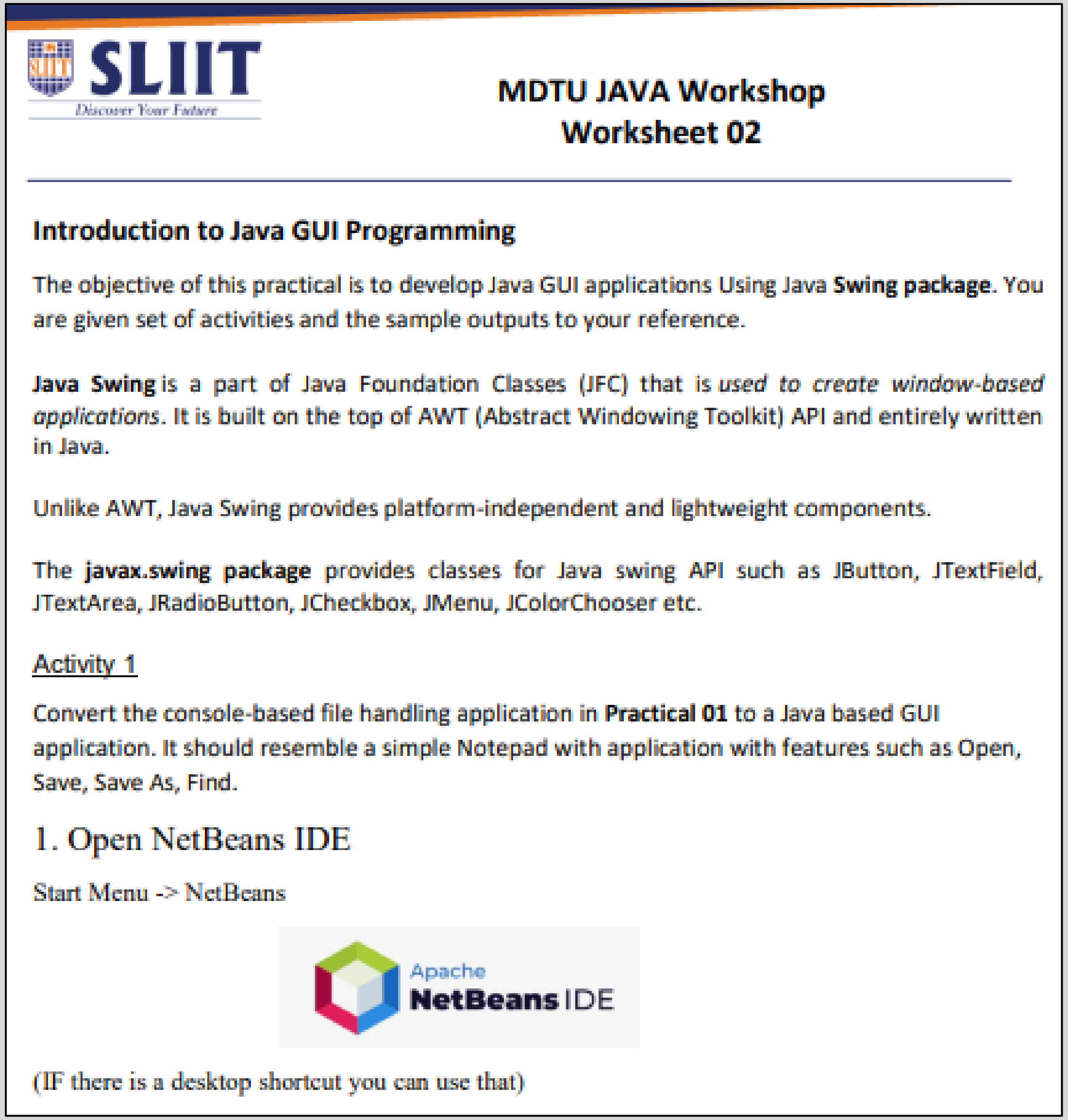
- Task – take one sentence as input from the user, and:
- (a) count how many words
- (b) count how many characters
- (c) find a character at a given position
- (d) remove all occurrences of a given character
- (e) find if a particular word is included in the sentence – if so, give details

Java GUI Programming

- Creating a Graphical Java Project on NetBeans
- GUI containers, components and properties
- Using NetBeans Palette and Properties
- Responding to Events – ActionEvents, MouseEvents, KeyEvents, etc.
- GUI Data Models – e.g., Jlist and Vector
- Calling GUIs from one another – and transferring data between them
- (!!) GUIs and Multithreading
- (!!) Graphic Programming using Canvas and Java2D

Practical 2

- Convert the console-based file handling application done in Practical 1 to a Java based GUI application.
- Application should resemble a simple Notepad application which has all the features of the previous console application, but also act as an editor for simple text files
- Include Features such as **New**, **Open**, **Save**, **Save-As**, and other additional features like **Find**, **Find Next**, **Replace**, printing line counts, word counts, character counts…



SLIIT
Discover Your Future

MDTU JAVA Workshop
Worksheet 02

Introduction to Java GUI Programming

The objective of this practical is to develop Java GUI applications Using Java **Swing package**. You are given set of activities and the sample outputs to your reference.

Java Swing is a part of Java Foundation Classes (JFC) that is *used to create window-based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in Java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The **javax.swing package** provides classes for Java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

Activity 1

Convert the console-based file handling application in **Practical 01** to a Java based GUI application. It should resemble a simple Notepad with application with features such as Open, Save, Save As, Find.

1. Open NetBeans IDE

Start Menu -> NetBeans

(IF there is a desktop shortcut you can use that)

What did we learn today?

- Introduction to Java
- Introduction to NetBeans
- Java Programming Essentials
- Object Oriented Programming Basics
- File Handling
- Building console applications

JAVA WORKSHOP

Day 3 – 02/12

What Did We Learn Yesterday

- Check previous days homework

Methods

- Methods can be used to re-use duplicate code
- **Duplicate is the root of all evil!**
- Refactoring is a best practice in the industry to remove duplicate code from time to time, by collecting the duplicate code into methods, or sometimes, classes.

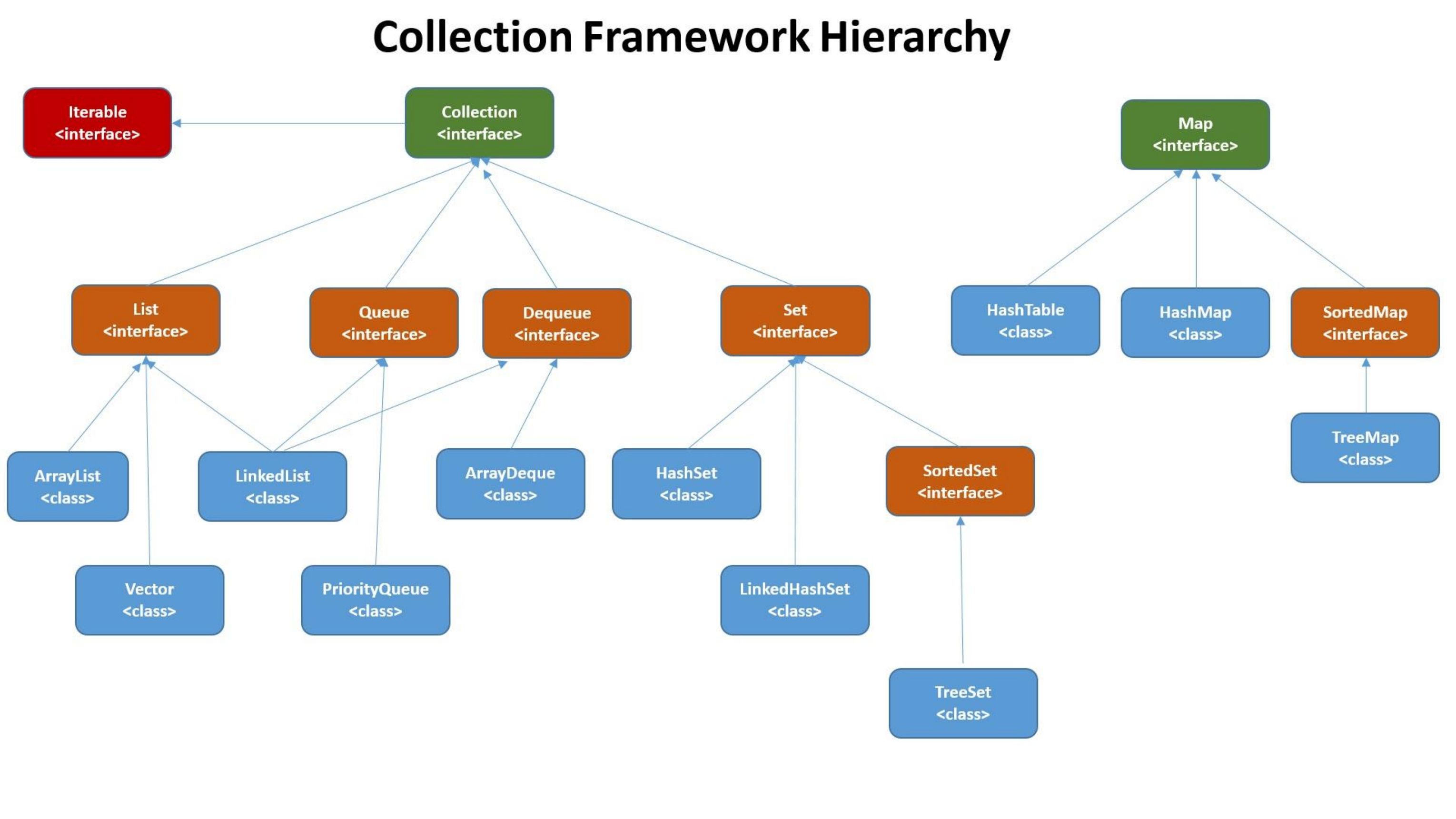
- Methods have a method signature, argument list, and return type

Exceptions

- To handle foreseen (and sometimes, unforeseen) problems
- Specified using
  – try, catch, Exception keywords
  - throws keyword
- Mostly used in file I/O and database operations
- Exceptions must always be handled properly, otherwise your users will be very unhappy!

Arrays & Collections

- Used to store multiple similar type variables or objects.
- Arrays are faster, but have a fixed type and fixed length
- Collections are more flexible, dynamic sized, and could have generic types



Threads

- This is so that the GUI (graphical user interface) does not freeze, or become unresponsive during a blocking operation
- Blocking operations are either CPU intensive or IO intensive operations
- CPU intensive – computing a result
- IO intensive – reading a file over the network
- To use multithreading, each platform provides different methods
    - General – Runnable.run()
    - Swing – SwingUtilities.invokeLater()
    - JavaFX – Platform.runLater(), etc.

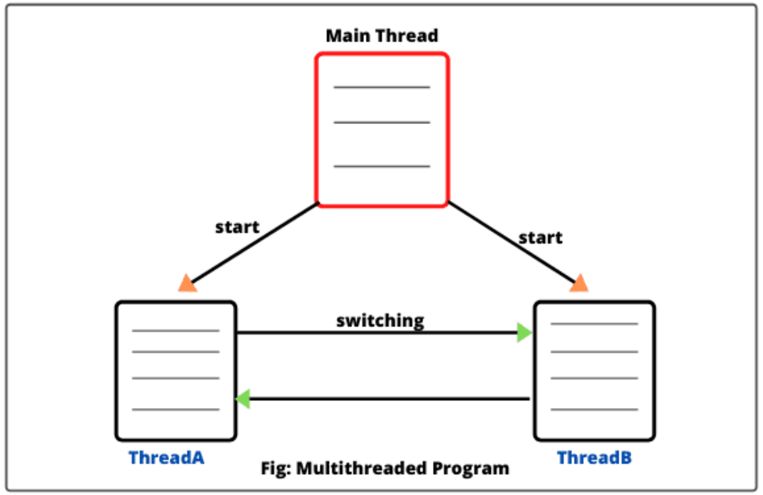
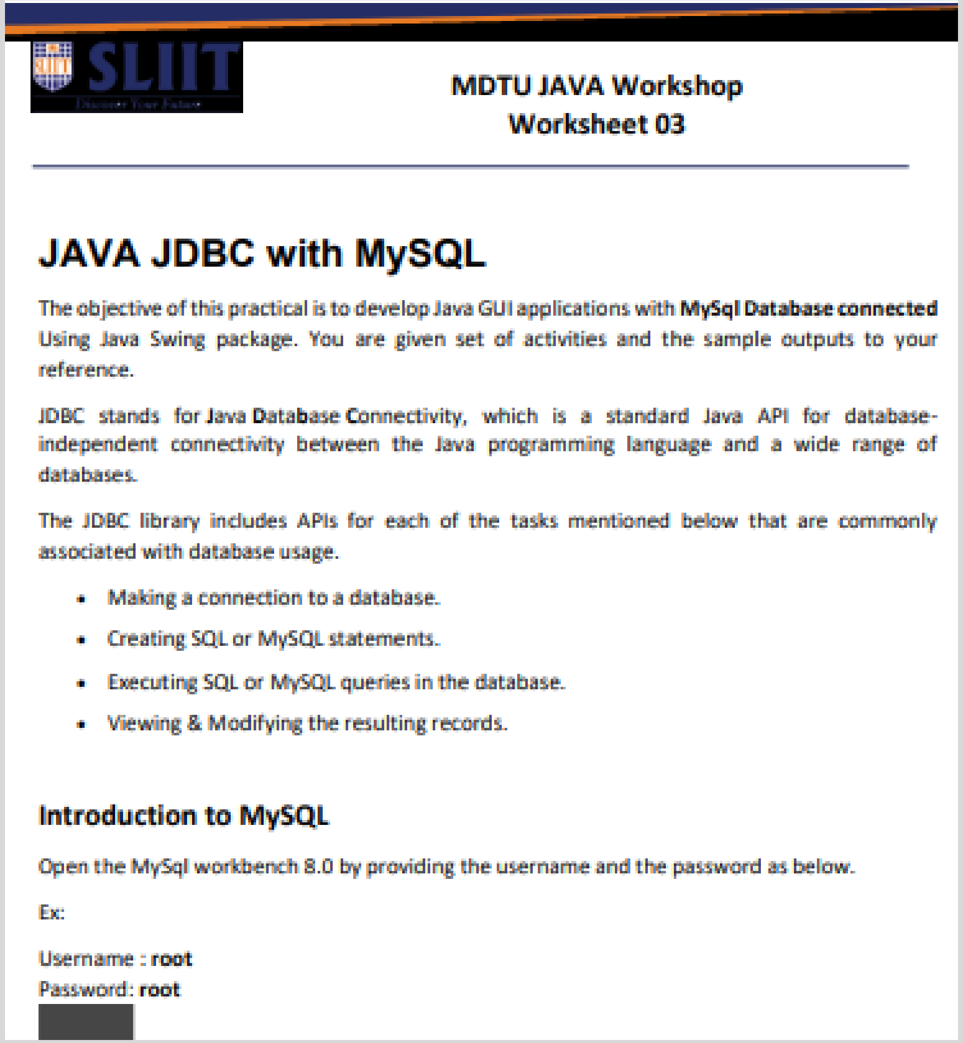


Fig: Multithreaded Program

Practical 3

- **Practical First**
- Refer to the practical sheet for instructions on how to setup MySQL and connect with JDBC and perform essential operations
- Connect to DB
- Write to DB
- Read from DB
- We will do other examples in the classroom afterwards



SLIIT

MDTU JAVA Workshop
Worksheet 03

JAVA JDBC with MySQL

The objective of this practical is to develop Java GUI applications with **MySql Database connected** Using Java Swing package. You are given set of activities and the sample outputs to your reference.

JDBC stands for Java Database Connectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

The JDBC library includes APIs for each of the tasks mentioned below that are commonly associated with database usage.

- Making a connection to a database.
- Creating SQL or MySQL statements.
- Executing SQL or MySQL queries in the database.
- Viewing & Modifying the resulting records.

Introduction to MySQL

Open the MySql workbench 8.0 by providing the username and the password as below.

Ex:

Username : **root**
Password: **root**

In Class Exercise (3pm-5pm)

- This is a slightly simplified version of the previous story
  - Create MySQL tables and populate with the relevant data.
  - Create a Java GUI application and connect using JDBC.
  - Provide necessary functions to perform CRUD operations.

- A student has a name, student id number, and age. Teachers have a name and employee number. A module has a module code, a teacher and students. John and Jane are two students. Smith is a teacher. Dan is a teacher. John's student id is 1, Jane's student id is 2. John is 21 years old. Jane is 23 years old. Smith has an employee number of 1000. Dan has an employee number of 1001. Smith teaches Programming. Dan teaches Graphic Designing. John and Jane take Graphic Designing. John also takes Programming.

JAVA WORKSHOP

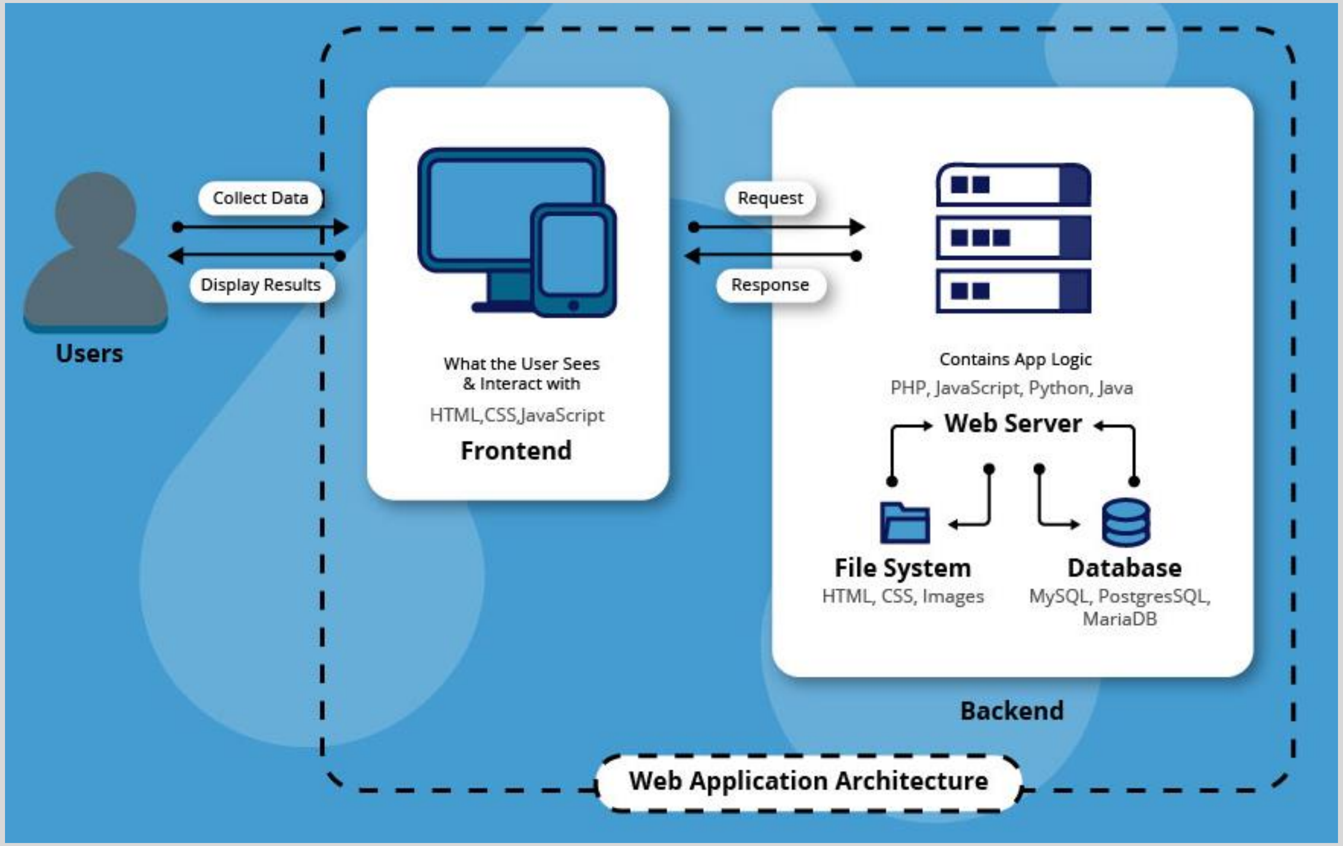
Day 4 – 03/12

What Did We Learn Yesterday

- Check previous days homework

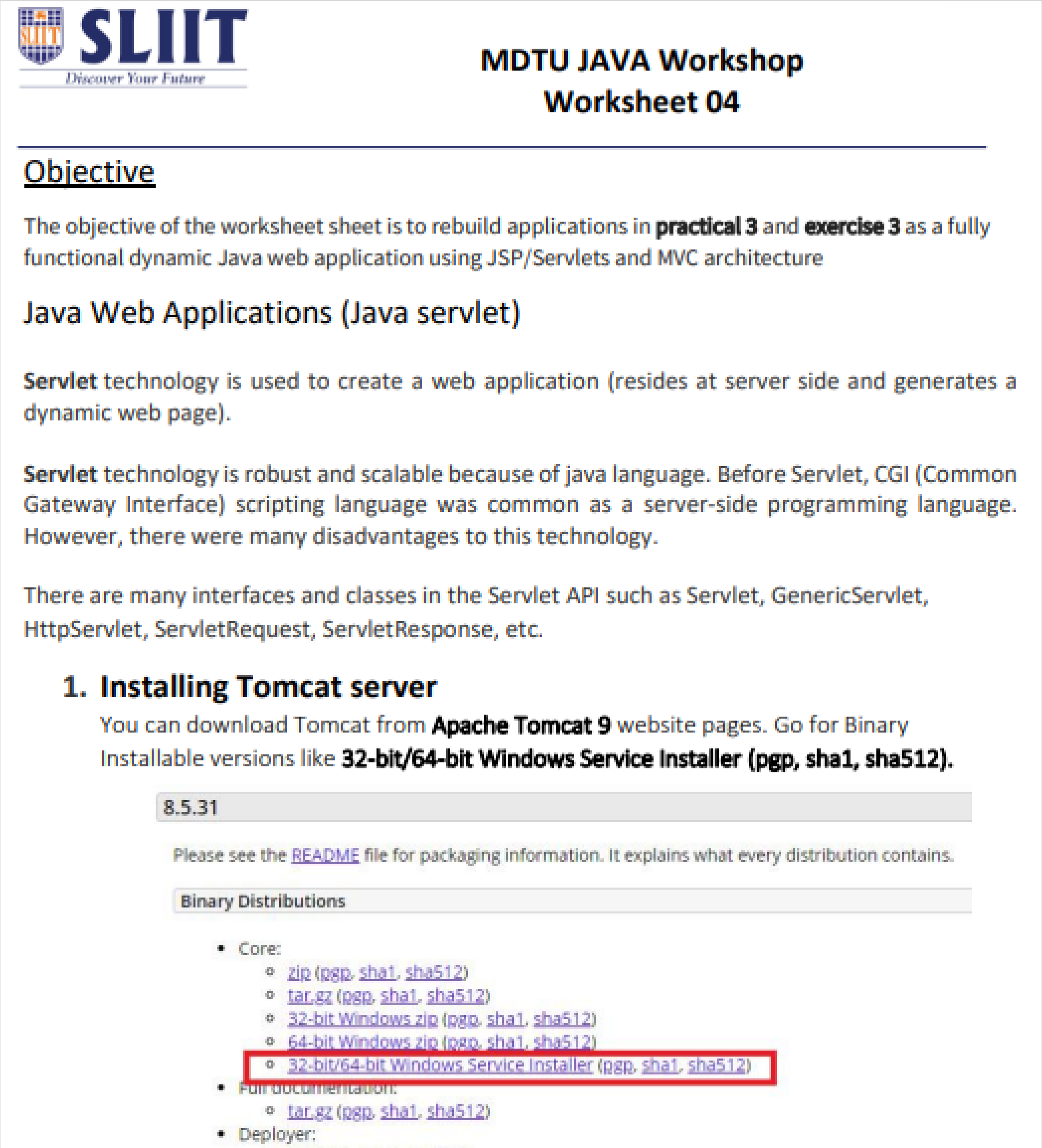
Java Web Applications

- HTTP protocol
- Static vs dynamic web pages
- Java web application basics
- Application servers
- Installing and Configuring Tomcat.
- What are JSPs
- What are Servlets
- Deploying web applications,
- Deployment Descriptor (XML)
- Connecting Tomcat with JDBC
- MVC Architecture
- (!!) Filters,
- (!!) Listeners
- (!!) Tag Libraries



Practical 4

- Rebuild application in practical 3 as a fully functional dynamic Java web application using JSP/Servlets and MVC architecture.
- Start from creating a new Java Web Project using NetBeans
- How to setup the DB for this, and connect the DB, using NetBeans
- Application should be like a simple CRUD application like the practical 3



SLIIT
Discover Your Future

MDTU JAVA Workshop
Worksheet 04

Objective

The objective of the worksheet sheet is to rebuild applications in **practical 3** and **exercise 3** as a fully functional dynamic Java web application using JSP/Servlets and MVC architecture

Java Web Applications (Java servlet)

Servlet technology is used to create a web application (resides at server side and generates a dynamic web page).

Servlet technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology.

There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

1. Installing Tomcat server

You can download Tomcat from **Apache Tomcat 9** website pages. Go for Binary Installable versions like **32-bit/64-bit Windows Service Installer (pgp, sha1, sha512).**

8.5.31

Please see the README file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
  - zip (pgp, sha1, sha512)
  - tar.gz (pgp, sha1, sha512)
  - 32-bit Windows zip (pgp, sha1, sha512)
  - 64-bit Windows zip (pgp, sha1, sha512)
  - 32-bit/64-bit Windows Service Installer (pgp, sha1, sha512)
- Full documentation:
  - tar.gz (pgp, sha1, sha512)
- Deployer:

JAVA WORKSHOP

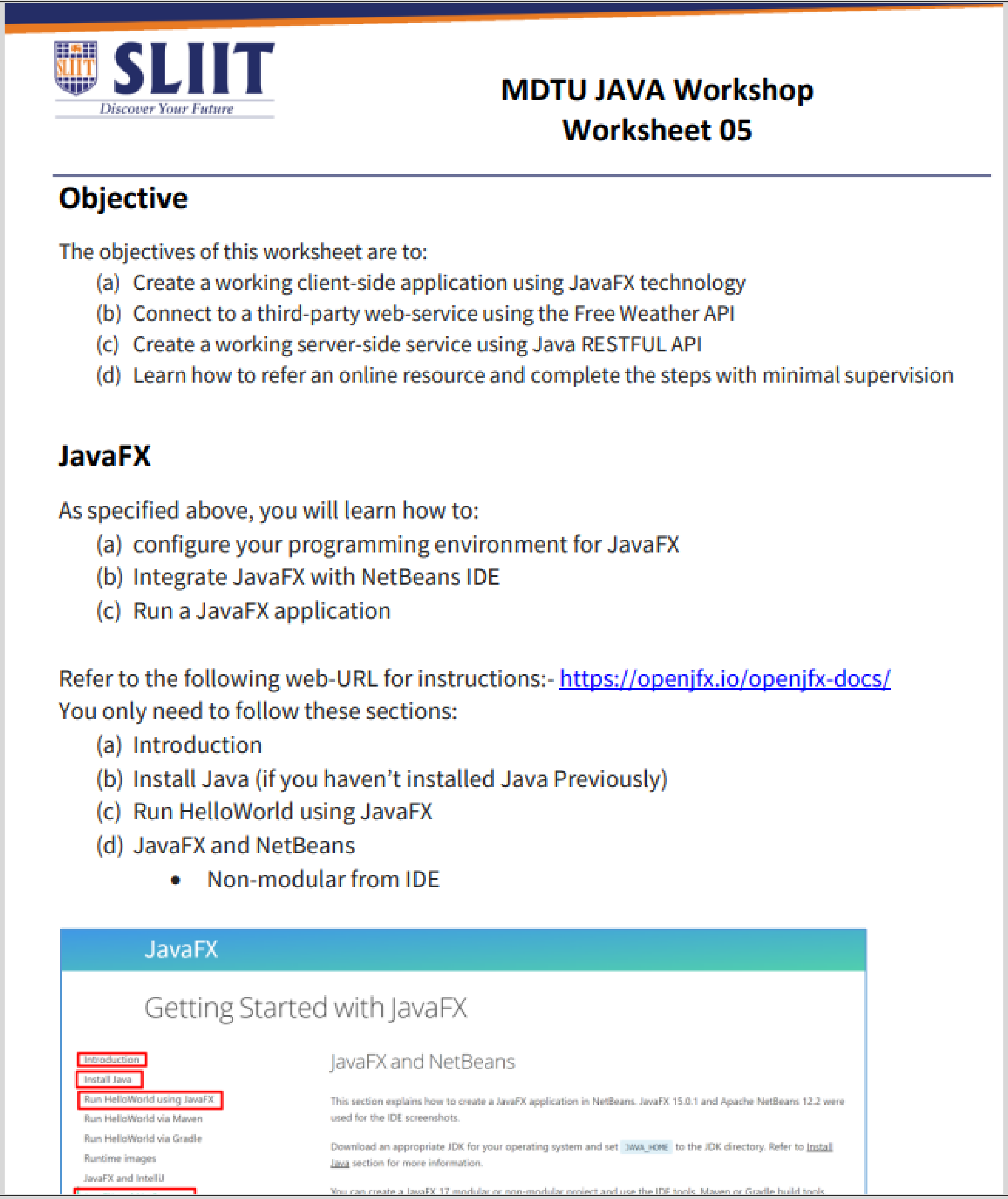
Day 5 - 04/12

What Did We Learn Yesterday

- Check previous days homework

Practical 5

- Develop a Java Enterprise Application
- Two Backends
  - Third-party backend (free weather API)
  - Our own backend (using Java-REST)
- Three Frontends
  - Web browser
  - JavaFX
  - Simple HTML Page with AJAX/jQuery
- Exercise:
  - We will try to do our practical by referring to a web resource with minimal supervision



SLIIT
Discover Your Future

MDTU JAVA Workshop
Worksheet 05

Objective

The objectives of this worksheet are to:

(a) Create a working client-side application using JavaFX technology
(b) Connect to a third-party web-service using the Free Weather API
(c) Create a working server-side service using Java RESTFUL API
(d) Learn how to refer an online resource and complete the steps with minimal supervision

JavaFX

As specified above, you will learn how to:

(a) configure your programming environment for JavaFX
(b) Integrate JavaFX with NetBeans IDE
(c) Run a JavaFX application

Refer to the following web-URL for instructions:- https://openjfx.io/openjfx-docs/
You only need to follow these sections:

(a) Introduction
(b) Install Java (if you haven't installed Java Previously)
(c) Run HelloWorld using JavaFX
(d) JavaFX and NetBeans
  - Non-modular from IDE

JavaFX

Getting Started with JavaFX

- Introduction
- Install Java
- Run HelloWorld using JavaFX
- Run HelloWorld via Maven
- Run HelloWorld via Gradle
- Runtime images
- JavaFX and IntelliJ

JavaFX and NetBeans

This section explains how to create a JavaFX application in NetBeans. JavaFX 15.0.1 and Apache NetBeans 12.2 were used for the IDE screenshots.

Download an appropriate JDK for your operating system and set `JAVA_HOME` to the JDK directory. Refer to Install Java section for more information.

You can create a JavaFX 17 modular or non-modular project and use the IDE tools, Maven or Gradle build tools