



**Java Institute for Advanced Technology**

**UNIT NAME:** SOFTWARE APPLICATION DEVELOPMENT

**UNIT ID:** H7E1 04

**ASSIGNMENT NAME:** RESEARCH ASSESSMENT

**NAME:** M.R.P.N. THARUKSHA RAJAPAKSHA

**STUDENT ID:** 2019/2020/CO/SE/I2/029

**SCN NO:** 207977608

**NIC:** 200019401866

**BRANCH:** JAVA INSTITUTE, COLOMBO



## Section A

1. What is the long form of API?
  - a. Access protocol Interface
  - b. Application Programming Interface
  - c. Access Programming Interface
  - d. Application Programming Internet
2. What is the correct statement from given below?
  - a. Javax.swing package introduced prior to javax.awt package
  - b. Java.awt package introduced prior to javax.awt package
  - c. Java.awt package introduced prior to javax.swing package
  - d. Answer not given
3. Which package include jTextField object with this constructor „new jTextField()“?
  - a. Java.sql
  - b. Java.util
  - c. Javax.swing
  - d. com.date
4. What is the suitable method for clear jTextField ?
  - a. fieldname.setText(“delete”);
  - b. fieldname.setValue(“”);
  - c. fieldname.setData(“”);
  - d. fieldname.setText(“”);
5. What is the Correct JDBC Driver name for MySQL 8+
  - a. com.mysql.jdbc.Driver
  - b. com.mysql.cj.jdbc.Driver
  - c. Com.Mysql.Jdbc.Driver
  - d. Jdbc:mysql://
6. What is the return type in executeUpdate(“Select \* FROM Products”)
  - a. boolean
  - b. int
  - c. ResultSet
  - d. Answer not Gievn
7. Which method can save data from database?
  - a. getData(sql);
  - b. execute(sql);
  - c. executeQuery(sql);
  - d. executeUpdate(sql);

8. Which method can change size a JFrame?
- a. setHightAndWidth();
  - b. setFrameDimension();
  - c. setFrameSize();
  - d. Answer not Given
9. What is the parameter data type which use in execute ();
- a. int
  - b. double
  - c. String
  - d. boolean
10. What are the data types which can use for add in to an ArrayList?
- a. Primitive data
  - b. Non-primitive data
  - c. Both primitive and non-primitive data
  - d. Int
11. actionPerformed() method helps to detect \_\_\_\_\_
- a. an Event
  - b. colors
  - c. sounds
  - d. Errors
12. What is the correct statement based on the Image?



- a. The Letter A pointed to textField
  - b. The Letter A pointed to Menu
  - c. The Letter A pointed to MenuItem
  - d. The letter A pointed to Label
13. What is the most suitable parameter for JLabel class Constructor?
- a. flot
  - b. String
  - c. int
  - d. Double

14. What is the return type for getPassword() which defined at jPasswordField?
- a. String
  - b. String[]
  - c. char[]
  - d. int
15. What are the steps for set an Image to JLabel?
- a. Label > Properties > image
  - b. Label > Customcode > imageurl
  - c. Label > Properties > icon
  - d. Label > Properties > imageprperty
16. EXE4J application used for?
- a. Generate an EXE file for Windows Computers
  - b. Generate EXE file for Linux Computers
  - c. Generate EXE file for Mac Computers
  - d. None of above
17. Which component guarantees that only one button can be selected at a time?
- a. RadioButton
  - b. JRadioButton
  - c. ButtonGroup
  - d. jCheckBox
18. What is the output here?
- ```
String x="455456.44";  
int u = Integer.parseInt(x);  
System.out.println(u);
```
- a. 455456
  - b. 455457
  - c. Compile Error
  - d. Run Time Error
19. Assume you want to get highest value from a numeric data set. Select the suitable method to fill following code fragment.
- ```
ResultSet rs = DB.getCon().CreateStatement().executeQuery("_____?_____");
```
- a. SELECT Celing(column\_name) from tableName
  - b. SELECT MAX(column\_name) from tableName
  - c. SELECT MIN(column\_name) from tableName
  - d. Answer Not Given

20. Consider the following code;

```
private void tf1KeyTyped(java.awt.event.KeyEvent evt) {  
    int x= evt.getKeyCode();  
    if(c==10){  
        System.out.println("OK");  
    }  
}
```

Which is the result of attempting to compile and run the above netbeans event ?

- a. Print : OK after type any key
- b. **Print Nothing**
- c. Print OK after type enter button
- d. Print OK after type “,” (comma)

21. What is the correct Logger Level from the given below?

- a. DEFRAG
- b. **FATAL**
- c. EXCEPTION
- d. DEFEAT

22. JasperCompileManager.compileReport() method return type is?

- a. JasperFillReport
- b. JasperPrint
- c. **JasperReport**
- d. None of these

23. What is the incorrect Core object according to log4j Architechure

- a. Logger Object
- b. Layout Object
- c. Appender Object
- d. Layout Object

**24. void assertTrue(boolean condition)**

- a. used for financial calculations, calculate worth
- b. used for testing, checks that a condition is false
- c. **used for testing, checks that a condition is true**
- d. Answer not given

25. What is the appropriate data type for store set of data, which came from a database or a Table in Jasper Reports?

- a. **Fields**
- b. Variables
- c. Parameters
- d. Answer Not given

## Section B

1. “API Documentation is the manual, containing all the information required to work with the corresponding source code” Do you agree with the above sentence? Evaluate your answer. (250 Words)

Yes, I agree with the statement "API Documentation is the manual, containing all the information required to work with the corresponding source code". Because simply put, API documentation contains how to properly handle a source code. It is a document like a user manual. For example, a document that gives developers a complete understanding of how to handle items such as classes, methods, objects, variables in a java source code (Java API).

Explained in more detail, API documentation is a technical content deliverable, containing instructions about how to effectively use and integrate with a source code. It's a concise reference manual containing all the information required to work with the source code, with details about the functions, classes, methods, objects, variables, return types, data types, arguments, and more, supported by tutorials and examples. API documentation is traditionally done using standard content creation and maintenance tools and text editors.

Developers are the ones who benefit the most from this API documentation. This is because they use various source codes to develop their software development and programming. They need knowledge of several source codes in their work. Source codes have a number of different features. Also, a single source code contains a large number of functions, classes, methods, objects, variables, return types, data types, and arguments. In practice, a developer cannot remember all this. Therefore, having API Documentation for source codes allows developers to gain an understanding of how to manipulate the items they need when they need them. Also, this API documentation contains all the relevant information so there is no need to search elsewhere. Therefore, the work of developers is very easy and efficient.

Finally, API Documentation leads to better product maintenance, speeding up maintenance and updates.

2. There are many types of barcodes—and each brings unique opportunities and limitations. Should you invest in 1D or 2D barcodes? Explain your answer. (250 Words)

Nowadays almost all businesses tend to use barcodes to make their work easier and more efficient. This is because the use of barcodes does not require much knowledge, the encrypted data can be scaled to suit the needs, the competitive business environment can be facilitated, barcodes can be mounted on any surface required, reducing manpower and training time and thereby increasing productivity. error reduction, increase speed, time-saving, barcodes provide asset & security tracking, theft deterrence, peace of mind and significant reduction of losses/liabilities, cost-effectiveness, and etc.

There are many types of barcodes. 1D and 2D Barcodes are the two most commonly used Barcodes. However, each of these barcodes has unique opportunities and limitations. Depending on those unique opportunities and limitations, the appropriate barcode type should be used.

### **1D Barcodes**

These are also known as linear barcodes. A 1D barcode is a visible black and white pattern that uses space to encode variable width lines and information. That information - like numbers or other keyboard features - is encrypted horizontally from left to right. 1D barcodes have a limited number of characters, usually 20-25. To add more numbers, the barcode must be longer. 1D barcodes, however, do not contain complex digital data and can only provide tokens for text and number codes. A 1D barcode depends on database connectivity in order to be meaningful; After a scanner reads the numbers in the code, they must be reconnected to the product or pricing date.

1D barcode scanner technology is simpler, more affordable, and faster because of how simple the code is. Print quality does not have to be as sharp as some modern 2D codes. Integration with 1D barcode software is long-established and generally easy to work with. The 1D scanner has strong motion tolerance and long-range. Training for 1D scanners is extremely minimal because they are not overly sensitive and have limited capabilities. The code often works even when it is damaged.

However, access to an external database is required to decode their meaning. This is often the case even when newer versions of 1D software are deployed. The data storage capacity of these barcodes is severely limited. New scanners using optical recognition are notorious for having a hard time scanning these quickly.

The most familiar 1D barcodes are the common UPC codes found in grocery and consumer goods.

### **2D Barcodes**

2D barcodes use a grid of black and white pixels to create geometric patterns such as squares and hexagons. A 2D barcode uses patterns, shapes, and dots to encode information horizontally (X) and vertically (Y). The 2D barcode can encode more characters (about 2000) in as much space as the 1D barcode. The process takes up less physical space and 2D barcodes can be very dense. Binary information can also be stored, albeit with limited scope, with alphanumeric data.

2D codes include QR code, PDF417, and Data Matrix. In addition to retaining a large amount of data, 2D barcodes can encrypt images, web addresses, and other binary data, which means that the codes can be executed independently of a database.

2D barcodes take up less than the amount of data that can be stored. These barcodes can be read from any angle. They can store some local data, which reduces the need for wireless communication when scanning them. They can store more types of data and, among other things, can add to security if the data is encrypted on top of being encoded. Many of these can be scanned by a large range of devices. And these can be scanned on any phone with a good camera.

However, these can be difficult to read if you look closely at some 2D barcodes. They often require a sharper printer to print them, rather than lasers. Scanning these barcodes can be slow. Because of the slow visual nature of the hardware, users need some training and a steady, patient hand.

The most commonly used 2D barcode is the QR (Quick Response) code. This code is very popular because of its fast readability and ability to store large amounts of information.

1D and 2D barcodes have their own strengths and weaknesses. Functionally, the crucial difference between the two types of barcodes lies in the type of scanner they need to read. 1D code can be scanned with a traditional laser scanner. 2D barcodes should be read using a scan engine called imager.

1D code data is used in cases where data is subject to change. 1D barcodes are also useful for providing limited-length tracking information for faster database searches. 2D barcodes are used where there is no database connection, space is limited and a large amount of data is required. 2D barcodes are also the best option for those looking for a fast and accurate barcode read type. This scanning technology has increased the 2D barcodes used for manufacturing and supply chain applications as they efficiently scan moving items on the conveyor belts without having to worry about scanner alignment.

1D and 2D barcodes are best suited for different situations. Want to make scanning as fast and easy as possible for a cashier? Technology does not age well and may be stubborn to run, but you definitely need 1D barcodes. Are you an event manager, looking to facilitate access but your staff does not have access to gun scans? 2D barcodes that can be read by smartphones like QR codes make more sense to you. Functionally, what type of code is used depends on the application. But many companies include both 1D and 2D barcodes in their labels at the request of customers or in accordance with industry-specific labeling standards.

Knowing the unique opportunities and limitations of both 1D and 2D barcodes will allow you to decide on the best barcode type for your business. Accordingly, you can invest in the most suitable type of barcode.



3. In case of problems with an application, programmers want to enable logging so that the problem can be located. There are some pre developed tools which helps to archive this requirement. Introduce one of them and explain its implementation process with a sample code fragment. (250 Words)

In case of problems with an application, programmers want to enable logging so that the problem can be located. When logging in to an application is enabled, the system keeps a record of the errors and abnormalities that occur when the user of the application does something with the application and what happens in the application. Experience shows that logging was an important part of the development cycle. It offers several advantages. It provides an accurate context for running the app. Once entered into the code, no human intervention is required to generate the logging output. Moreover, the log output can be saved to a stable medium for later study. In addition to its use during the development cycle, a sufficiently rich logging package can also be considered as an audit tool. Adding log statements to the code is a less technical way of debugging it. It may be the only method because debuggers are not always available or relevant. This is usually the case for multithreaded applications and widely distributed applications.

There are some pre-developed tools that help to archive this requirement. The most commonly used pre-developed tool is the API provided through the `org.apache.log4j.*` Package. This is used for Java.

### **log4j**

Almost every major application has its own logging or tracing API. In accordance with this law, the E.U. The SEMPER project decided to write its own discovery API. This was in early 1996. After countless enhancements, several incarnations, and much more, the API has evolved into log4j, a popular login package for Java. The package is distributed under the Apache Software License, a fully open source license certified by Open Source Initiation. This login package allows the developer to create logs, move logs to the appropriate destination, and maintain a specific format.

There are flaws in logging. It can slow down an application. If too verbose, it can cause scrolling blindness. To alleviate these problems, log4j is designed to be reliable, fast, and extensible. Since logging is rarely the main focus of an application, the log4j API strives to be simple to understand and to use.

Each logger machine keeps a record of the closest existing ancestors in the logger namespace and also has a related "level". Log levels are used to classify logs according to their severity or impact on application stability. The `Org.apache.log4j.*` Package also provides different levels of login.

**org.apache.log4j.\* package provides the following levels in descending order:**

FATAL

ERROR

WARN

INFO

DEBUG

Loggers in Java are objects which trigger log events, They are created and are called in the code of the application, where they generate Log Events before passing them to the next component which is an Appender. You can use multiple loggers in one class to respond to different events or to use the logger in the hierarchy. They are usually named using hierarchical dot-separated namespace. Also, all names in the logger must be based on the class or package name of the logged component.

**Example of Logging in Java using the org.apache.log4j.\* package:**

```
import org.apache.log4j.*;

public class Supplier {

    static Logger logger = Logger.getLogger(Supplier.class.getName());

    public static void main(String[] args) {

        logger.error("ERROR");

        logger.warn("WARNING");

        logger.fatal("FATAL");

        logger.debug("DEBUG");

        logger.info("INFO");

    }

}
```

import org.apache.log4j.\*; -> import org.apache.log4j.\* package to the class.

static Logger logger = Logger.getLogger(Supplier.class.getName()); -> get loggers int the class and assign them to logger variable.

## **Properties File of Log4j Package Sample Log4j Properties file:**

```
# Enable Root logger option
log4j.rootLogger=INFO, file, dout, DB

# Attach appenders to print file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=E:loglogging.log
log4j.appender.file.MaxFileSize=10MB
log4j.appender.file.MaxBackupIndex=5
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L
- %m%n

# Attach appenders to print on console
log4j.appender.dout=org.apache.log4j.ConsoleAppender
log4j.appender.dout.Target=System.out
log4j.appender.dout.layout=org.apache.log4j.PatternLayout
log4j.appender.dout.layout.ConversionPattern=%d{yyyy-MM-dd      HH:mm:ss}      %-5p
%c{1}:%L - %m%n

# Attach appenders to store in database
log4j.appender.DB=org.apache.log4j.jdbc.JDBCAppender
log4j.appender.DB.URL=jdbc:mysql://localhost:3306/myLogs
log4j.appender.DB.driver=com.mysql.jdbc.Driver
log4j.appender.DB.user=root
log4j.appender.DB.password=123
log4j.appender.DB.sql=INSERT INTO LOG VALUES('%x',now(),'%C','%p','%m')
log4j.appender.DB.layout=org.apache.log4j.PatternLayout
```

The Log4j properties file is created inside the src folder of the project.

log4j.appender.file=org.apache.log4j.RollingFileAppender - Prints all logs in a file.

log4j.appender.file.File=D:\log\logging.log - Specifies the log file location.

log4j.appender.file.MaxFileSize=10MB - Maximum size of the log file to 10MB.

log4j.appender.file.MaxBackupIndex=5 - Limits the number of backup files to 5.

log4j.appender.file.layout=org.apache.log4j.PatternLayout - Specifies the pattern in which logs will print to the log file.

log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L  
- %m%n - Sets the default conversion pattern.

log4j.appender.dout=org.apache.log4j.ConsoleAppender - Prints all logs in the console.

log4j.appender.dout.Target=System.out - Set the target to System.out.

log4j.appender.dout.layout=org.apache.log4j.PatternLayout - Specifies the pattern in which logs will print to on the console.

log4j.appender.dout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p  
%c{1}:%L - %m%n - Sets the default conversion pattern.

log4j.appender.DB=org.apache.log4j.jdbc.JDBCAppender - Store all logs in the database.

log4j.appender.DB.URL=jdbc:mysql://localhost:3306/myLogs - Specifies the database URL .

log4j.appender.DB.driver=com.mysql.jdbc.Driver - Specifies the MySQL database driver.

log4j.appender.DB.user=root - Set the username of the database connection.

log4j.appender.DB.password=123 - Set the password of the database connection.

log4j.appender.DB.sql=INSERT INTO LOG VALUES('%x',now(),'%C','%p','%m') - Create the query statement.

log4j.appender.DB.layout=org.apache.log4j.PatternLayout - Specifies the pattern in which logs will store in the database.

Loggers – Responsible for capturing log records and passing them to the corresponding Appender.

Appenders or Handlers – They are responsible for recording log events to a destination. Appenders format events with the help of Layouts, before sending outputs.

Layouts or Formatters – Responsible to determine how data looks when it appears in the log entry.