ASSIGNMENT NO: 1 (JAVA)

Title: Collections and Generics.

Aim: Design a system with the help of advance data structures in Java and enhance the system using collections and generics.

Objectives:

1 To study collections and generics in java

Apparatus:

Ubuntu 18.04
jdk 1.8.0_181
jre 1.8.0_181
Eclipse IDE

Questions:

1. What is Java Collections Framework? List out some benefits of Collections framework.
2. Which are the basic interfaces of Java Collections Framework?
3. What is difference between Enumeration and Iterator interface?
4. What is difference between Array and ArrayList? When will you use Array over ArrayList?
5. What are Generics in Java? How Generics works in Java ?

Aim : To design a system with the help of advance data structures in Java and enhance the system using collections and generics.

Questions

1. What is Java Collections Framework? List out some benefits of Collections Framework.

Ans. Java Collections Framework is a unified architecture for representing and manipulating collections. Collections Framework contain the following :

Interfaces — These are abstract datatypes that represent collections.

Implementations — These are concrete implementations of collection interfaces.

Algorithms — These are the methods that perform useful computations.

Benefits of Collections Framework

i) Reduces programming effort
ii) Increases programming speed and quality
iii) Allows interoperability among unrelated APIs.
iv) Reduces effort to learn and to use new APIs.
v) Reduces effort to design new APIs.
vi) Fosters software reuse.

2. Which are the basic interfaces of Java Collections Framework?

Ans. The Collections Framework define several core interfaces. The interfaces that underpin collections are summarized as follows

Interface — Description

i) Collection — Enables you to work with groups of objects.

ii) Deque — Extends Queue to handle double-ended queue

iii) List — Extends Collection to handle sequences.

iv) NavigableSet — Extends SortedSet to handle retrieval of elements based on closest-match searches.

v) Queue — Extends Collection to handle special types of list in which elements are removed only from the head.

vi) Set — Extends Collection to handle sets, which must contain unique elements.

vii) SortedSet — Extends Set to handle sorted sets.

3. What is the difference between Enumeration and Iterator interface?

Ans.

Enumeration	Iterator
i) Using Enumeration, you can only traverse the collection. You can't do any modifications to the collection	i) Using Iterator, you can remove an element of the collection while traversing it.
ii) Enumeration is used to traverse legacy classes like Vector, Stack and HashTable.	ii) Iterator is used to iterate most of the classes in collections framework like ArrayList, HashSet, etc.
iii) Methods: hasMoreElements(), nextElement()	iii) Methos: hasNext(), next(), remove()
iv) Enumeration is fail-safe in nature.	iv) Iterator is fail-fast in nature.
v) Enumeration is not safe and secure due to its fail-safe nature.	v) Iterator is safe and secure.

4. What is the difference between Array and Arraylist?

Array	Arraylist
i) Array is a fixed length data structure.	i) Arraylist is a variable length Collection class.
ii) Generics can't be used along with an Array.	ii) Arraylist allows Generic type to ensure type-safety.
iii) length variable denotes length of Array.	iii) Arraylist provides size() method to calculate size of an Arraylist.
iv) Array can store both primitives and objects.	iv) Arraylist can store only objects.
v) Array use assignment operator to store the elements.	v) Arraylist use add() method to insert the elements.

5. What are Generics in Java? How Generics work in Java?

Ans. At its core, the term generics means parameterized types. Parameterized types are important because they enable you to create classes, interfaces and methods in which the type of data upon which they operate is specified as a parameter. Using generics, it is possible to create a single class, for eg., that automatically works with different types of data.

To implement Generics, the Java compiler applies Type Erasure to

i) Replace all type parameters in generic types with their bounds or Object if the type parameters are unbounded. The produced bytecode, therefore, contains only ordinary classes, interfaces, and methods.
ii) Insert type casts if necessary to preserve type safety.
iii) Generate bridge methods to preserve polymorphism in extended generic types.

Type Erasure ensures that no new classes are created for parameterized types; consequently, generics incur no runtime overhead.

ASSIGNMENT NO: 2 (Java)

TITLE: Client Server Architecture.

AIM: Enhance the system with the help of socket programming use client server architecture.

OBJECTIVES:

1. To do Client server connection in connection-oriented and connection Less Way.

OUTCOMES:

1. To apply appropriate advanced data structure and efficient algorithms to approach the problems of various domain.
2. To use effective and efficient data structures in solving various Computer Engineering domain problems.
3. To design the algorithms to solve the programming problems.

PRE-REQUISITES:

1. Knowledge Advance data structure.
2. Basic knowledge of Socket Programming.
3. Hands on Advance Java.

APPARATUS:

Ubuntu 18.04
jdk 1.8.0_181
jre 1.8.0_181
Eclipse IDE

Aim: To enhance the system with the help of socket programming. Use Client Server Architecture.

Questions

1. What is an IP Address?

Ans. An Internet Protocol address is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. An IP address serves two principal functions: host or network interface identification and location addressing.

2. What is Connection Oriented and Connection less Connection / Service?

Ans. Communication can be established in two ways between two or more devices that are connection-oriented and connection-less. Network layers can offer these two types of services to its predecessor layer for transferring data.

Connection-oriented — Analogous to the telephone system that requires communication entities to establish a connection before sending data. More precisely, it sets up a connection, uses that connection then terminates the

connection. Reliability is achieved by having recipient acknowledge each message. There are sequencing and flow control, that's the reason packets received at the receiving end are always in order. It uses circuit switching for transmission of data.

Connection-less — Connection-less service is analogous to postal system. Packets of data (usually known as datagram) is transmitted from source to destination directly. Each packet is treated as an individual entity, which allows communication entities to send data before establishing communication. Packets don't follow a fixed path that is the reason the packets received at receiving end can be out of order. It uses packet switching for transmission of data.

3 How to write Server Socket and Client Socket?

Ans. There are two types of TCP sockets in Java. One is for servers, and the other is for clients. The ServerSocket class is designed to be a listener, which waits for the clients to connect before doing anything.

When you create a ServerSocket, it will register itself with the system as having an interest in client connections.
The constructors for ServerSocket are
ServerSocket (int port) throws IOException
ServerSocket (int port, int maxQueue) throws IOException.
ServerSocket (int port, int maxQueue, InetAddress localAddress) throws IOException.

ServerSocket has a method called accept(), which is a blocking call that will wait for a client to initiate communications and then return with a normal Socket that is then used for communication with client.

The creation of a Socket object implicitly establishes a connection between the client and server. Here are two constructors used to create client sockets.
Socket (String hostName, int port) throws UnknownHostException, IOException.
Socket (InetAddress ipAddress, int port) throws IOException.

4. What is socket programming?

Ans. Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket listens on a particular port at an IP, while other socket

reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.

5. What is IOException?

Ans. public class IOException extends Exception.

Signals that an I/O exception of some sort has occurred. This class is the general class of exceptions produced by failed or interrupted I/O operations.

ASSIGNMENT NO: 3 (JAVA)

Title: JDBC

Aim: Enhance system by using JDBC, Multithreading, concurrency, synchronous and asynchronous callbacks, ThreadPools using ExecutorService.

Objectives:

1 To design system using JDBC
2 To design system using Multithreading concepts

Apparatus:

Ubuntu 18.04
jdk 1.8.0_181
jre 1.8.0_181
Eclipse IDE
mysql-connector-java-5.1.45-bin.jar

Questions:

1) What is JDBC?
2) Write the steps to connect to a database using JDBC application.
3) What is a Thread?
4) What is ThreadPool?
5) What is multithreading?

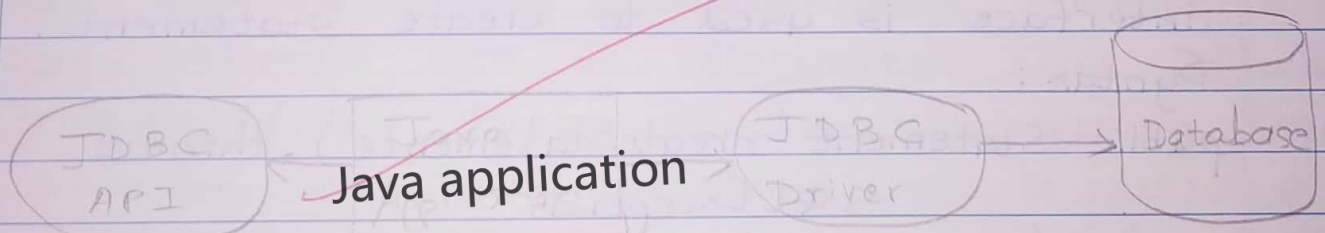
Aim : To design a system using JDBC.

Questions

1. What is JDBC?

Ans. Java DataBase Connectivity is an application programming interface for the JAVA programming language, which defines how a client may access a database. JDBC can perform following activities —

i) Connect to the database
ii) Execute queries
iii) Retrieve the result received from the database.



2. Write the steps to connect to a database using JDBC application.

Ans. There are five steps to connect any Java application with the database using JDBC.

i) Register the driver class
The forName() method of 'Class' class is used to register the driver class. This method is

used to dynamically load the driver class.
Syntax:
public static void forName(String className)
throws ClassNotFoundException.

ii) Create the connection object
The getConnection() method of DriverManager class is used to establish connection with the database.
Syntax:
public static Connection getConnection(String url, String name, String password) throws SQLException

iii) Create the Statement object.
The createStatement() method of Connection interface is used to create statement.
Syntax:
public Statement createStatement() throws SQLException.

iv) Execute the query
The executeQuery() is used to execute the queries to the database.
Syntax:
public ResultSet executeQuery(String sql)

v) Close the connection
The close() method is used to close the connection.
Syntax:
public void close() throws SQLException.

3. What is a thread?

Ans. A thread is a basic unit of CPU utilization; it comprises a thread ID, a program counter, a register set, and a stack. A thread, in the context of Java, is a separate path of execution.

4. What is Thread Pool?

Ans. If we allow all concurrent requests to be serviced in a new thread, we have not placed a bound on the number of threads concurrently active in the system. Unlimited threads could exhaust system resources, such as CPU time or memory. Also, knowing that creating a new thread will be discarded once it has completed its task is time consuming. One solution to these problems is to use a Thread Pool. The general idea behind a thread pool is to create a number of threads at process startup and place them into a pool, where they sit and wait for work.

5. What is multithreading?

Ans. Multithreading is the ability of an OS to support multiple, concurrent paths of execution within a single process.

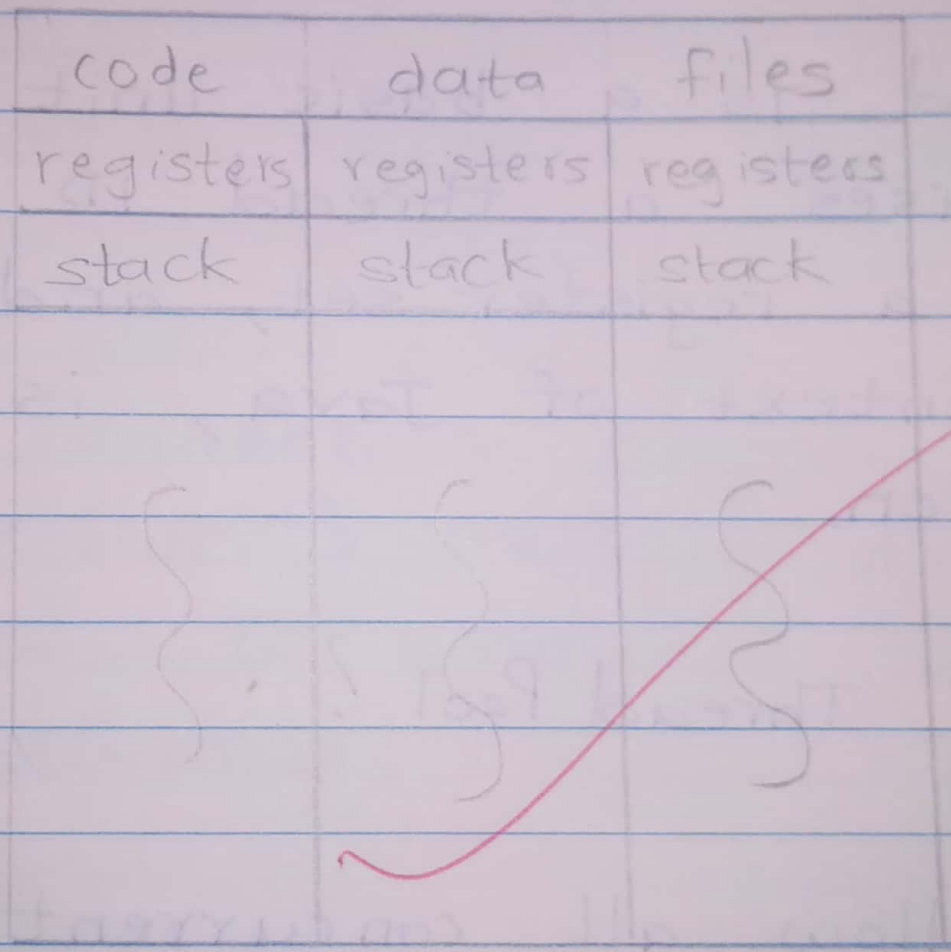


Fig. Multithreaded process

ASSIGNMENT NO: 4 (JAVA)

Title: Swing.

Aim: Transform the system from command line to GUI based application.

Objectives:

1. To study GUI in Java.
2. To study Functionality of AWT in Java.
3. To study Concept of Applet in Java.
4. To study Concept of Swing in Java.

Apparatus:

Ubuntu 18.04
jdk 1.8.0_181
jre 1.8.0_181
Eclipse IDE
mysql-connector-java-5.1.45-bin.jar

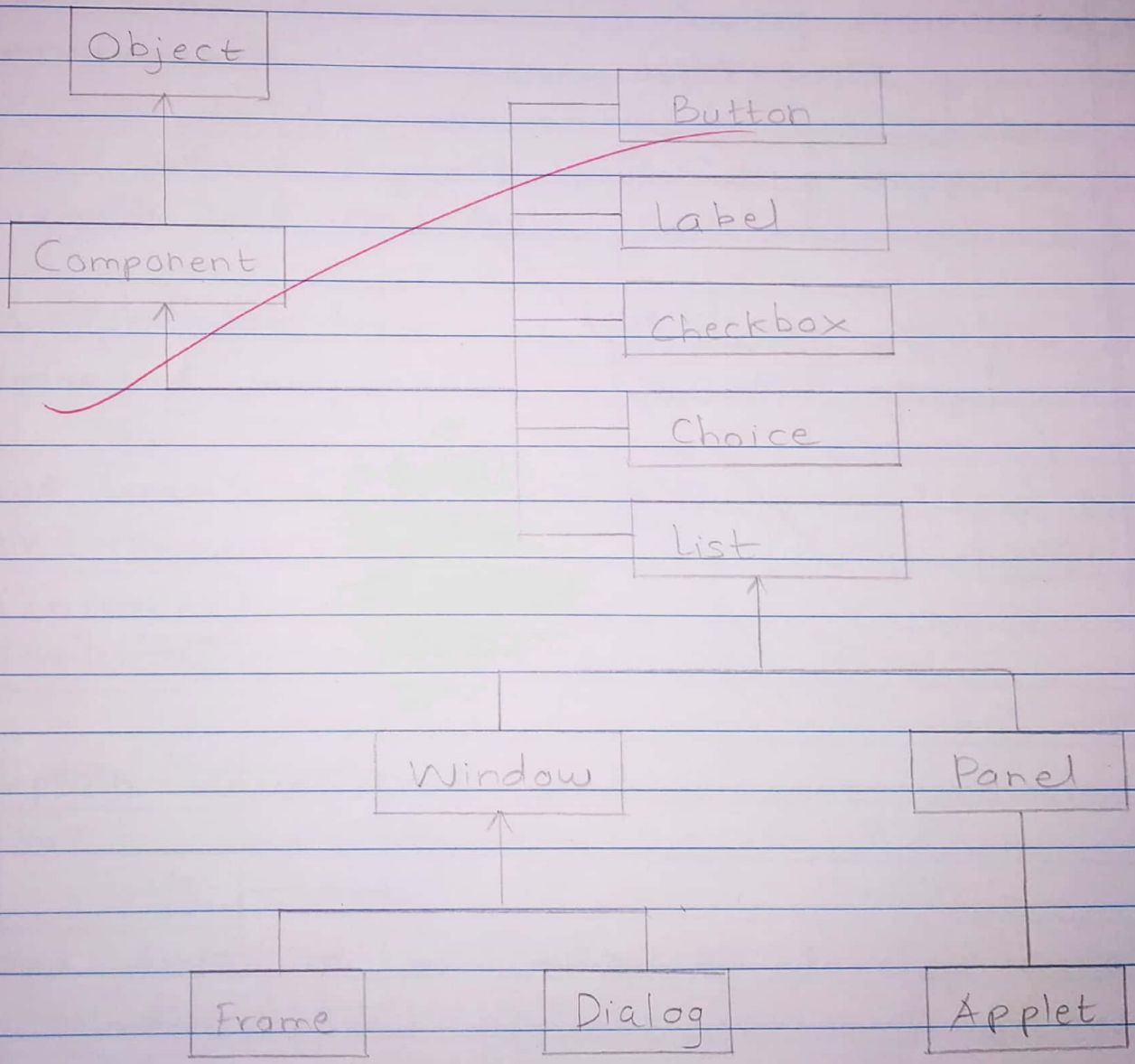
Questions:

1. Draw Java AWT Hierarchy.
2. Draw Hierarchy of Java Swing classes.
3. Difference between AWT and Swing.
4. Explain Java AWT and Classes provided by AWT.
5. Explain Life Cycle of Java Applet with methods.

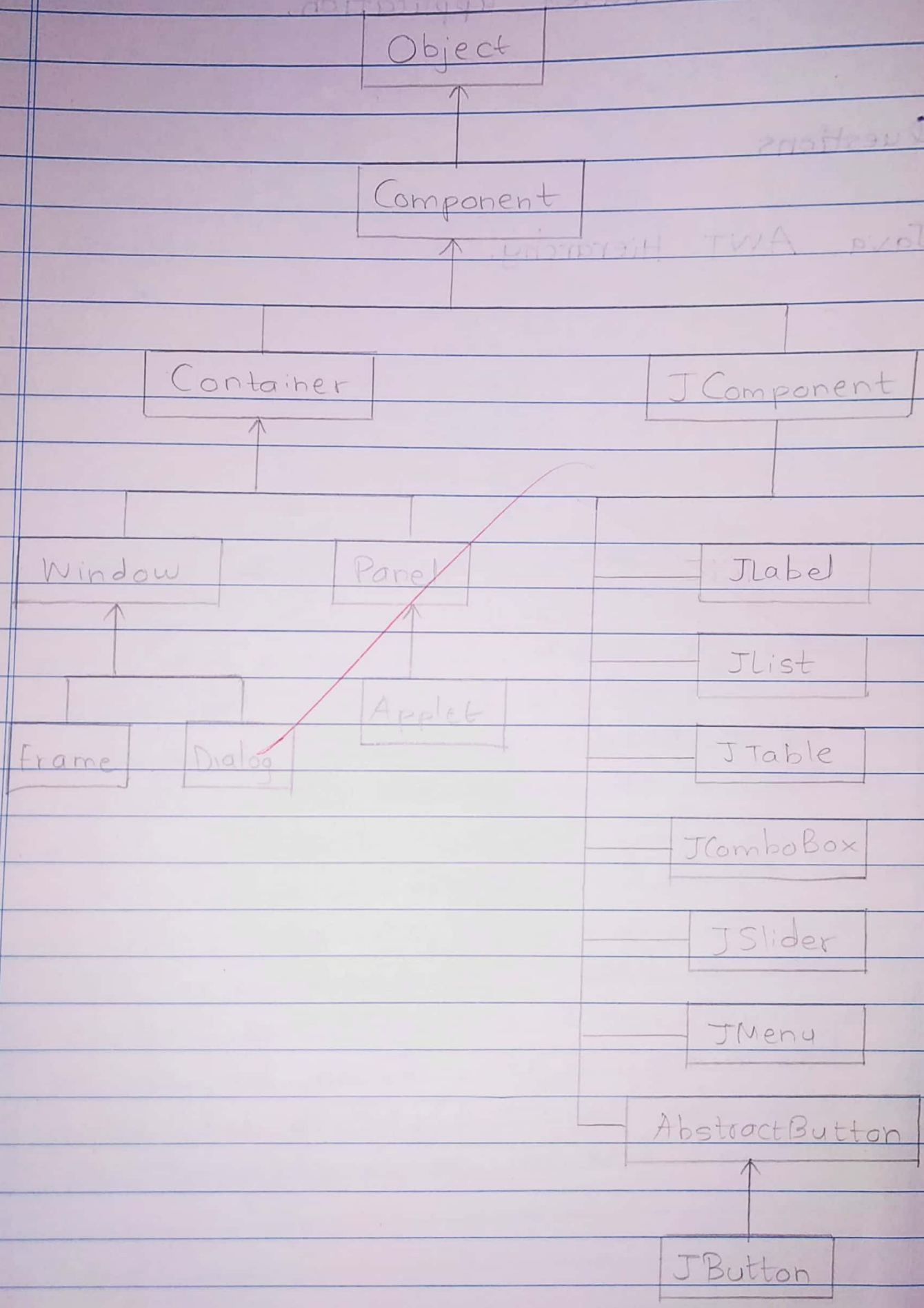
Aim : To transform command line system to GUI based application.

Questions

1. Java AWT Hierarchy.



2. Java Swing Hierarchy.



3. Difference between AWT and Swing.

AWT	Swing
i) AWT components are platform-dependent.	i) Swing components are platform-independent.
ii) AWT components are heavyweight.	ii) Swing components are lightweight.
iii) AWT doesn't support pluggable look and feel.	iii) Swing supports pluggable look and feel.
iv) AWT provides less number of components.	iv) Swing provides more powerful components.
v) AWT doesn't follow MVC (Model View Controller)	v) Swing follows MVC.

4. Explain Java AWT and classes provided by AWT.

Ans. Java AWT is an API to develop GUI or window based applications in Java. AWT is platform dependent because it calls native platform subroutine for creating components such as textbox, checkbox, button, etc. The AWT classes are contained in the

java.awt package.

Class — Description.

AWTEvent — Encapsulates AWT events.

AWTEventMulticaster — Dispatches events to multiple listeners

BorderLayout — The border layout manager.

Button — Creates a push button control.

Canvas — A blank, semantics-free window.

CardLayout — The card layout manager.

Checkbox — Creates a check box control.

CheckboxGroup — Creates a group of check box controls.

CheckboxMenuItem — Creates an on/off menu item.

Choice — Creates a pop-up list.

Color — Manages colors

Component — An abstract superclass for various AWT Components.

Container — A subclass of Component that can hold other components.

Cursor — Encapsulates a bitmapped cursor.

Dialog — Creates a dialog window.

Dimension — Specifies dimensions of an object.

EventQueue — Queues events

FileDialog — Creates a window from which a file can be selected.

FlowLayout — The flow layout manager.

Font — Encapsulates a type font

FontMetrics — Encapsulates various information related to a font.

Frame — Creates a standard window that has a title bar, resize corners, and a menu bar.

Graphics — Encapsulates the graphics context.

GraphicsDevice — Describes a graphics device such as a screen or printer.

GraphicsEnvironment — Describes the collection of available Font and GraphicsDevice objects.

GridBagConstraints — Defines various constraints relating to GridBagLayout class.

GridBagLayout — The grid bag layout manager

GridLayout — The grid layout manager.

Image — Encapsulates graphical images.

Insets — Encapsulates the borders of a container.

Label — Creates a label that displays a string.

List — Creates a list from which the user can choose.

MediaTracker — Manages media objects

Menu — Creates a pull down menu.

MenuComponent — An abstract class implemented by various menu classes.

MenuItem — Creates a menu item.

MenuShortcut — Encapsulates a keyboard shortcut for a menu item.

Panel — Simplest concrete subclass of Container.

Point — Encapsulates a Cartesian coordinate pair.

Polygon — Encapsulates a polygon.

PopupMenu — Encapsulates a pop-up menu.

PrintJob — An abstract class that represents a print job.

Rectangle — Encapsulates a rectangle.

Robot — Supports automated testing of AWT-based applications.

Scrollbar — Creates a scroll-bar control.

ScrollPane — A container that provides horizontal and/or vertical scroll bars for another component.

SystemColor — Contains the colors of GUI widgets.

TextArea — Creates a multiline edit control.

TextComponent — A superclass for TextArea and TextField.

TextField — Creates a single-line edit control.

Toolkit — Abstract class implemented by the AWT.

Window — Creates a window with no frame, no

menu bar, and no title.

5. Explain life cycle of Java Applet with methods.

Ans. Lifecycle of Java Applet

i) Applet is initialized
ii) Applet is started.
iii) Applet is painted.
iv) Applet is stopped
v) Applet is destroyed.

Lifecycle methods for Applet.

i) public void init() — is used to initialize the Applet. It is invoked only once.
ii) public void start() — is invoked after init() or browser is maximized. It is used to start the Applet.
iii) public void paint(Graphics g) — It is used to paint the Applet.
iv) public void stop() — is used to stop the Applet. It is invoked when Applet is to be stopped or browser is minimized.
v) public void destroy() — is used to destroy the Applet. It is invoked only once.

Singh