Name : Narendra Pyakurel
Roll no : 14
Subject : Java (advance)

1) Why Java is called platform independent language? What are JDK, JRE, JVM and Bytecode? Define how a Java source code file is executed using command mode, with example?

⇒ Java platform differ from most other platforms in the sense that it is a software based platform that runs on the top of other hardware based platforms. Java code can be run on multiple platforms. for ex. Windows, Linux etc. when Java code is complied into platform independent byte code and interpreted by the Java virtual machine on whichever platform it is being run on thats why Java is called platform independent.

JDK : The Java Development kit is a software development environment which is used to develop Java applications and applets. The JDK contains a private JVM and few other resources such as a compiler, interpreter

JRE : The Java Runtime Environment is a set of software tools which are used for developing Java application. It is used to provide the runtime environment. It contains a set of libraries and other files that JVM uses at runtime.

Scanned with CamScanner

JVM : The Java virtual machine is a specification that provides a runtime environment in which Java bytecode can be executed It can run those programs which are compiled Java byte code.

Bytecode : Java byte code is the result of the compilation of a Java program an intermediate representation of that program which is platform Independent. It gets processed by the JVM.

Executing Java source code using command mode.

lets say we have a class called A Java.

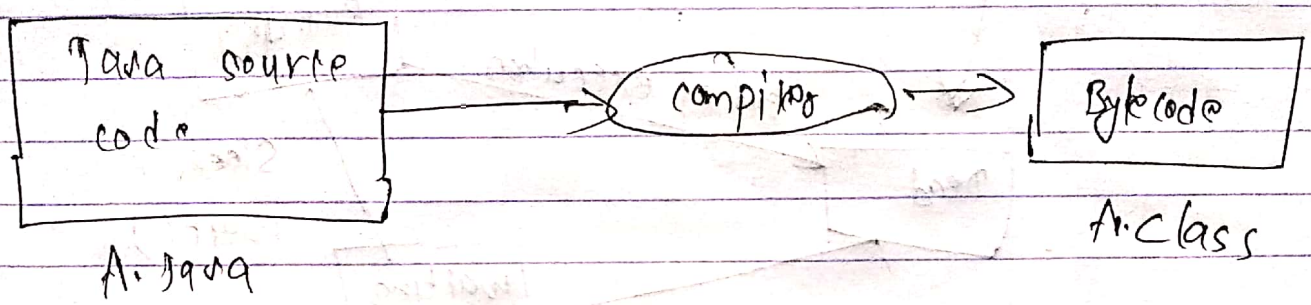
```
class A {

public static void main( string args [ ])
  {
   System.out.println(" hellow world")
```

To compile this Java file :

```
Javac A.Java
```

Scanned with CamScanner

At compile time, Java file is complied and converted into byte code



To execute A.class
Java A

Q.N.2) What is multithreading? Discuss life cycle of thread in detail, write different approach for multithreading.

⇒ The process of executing multiple threads simultaneously for maximum utilization of CPU is known as multithreading threads are light weight process within a process multithreading enables to write program in a way where multiple activities can proceed concurrently. It is mostly used in games and animation etc.

* Life cycle of thread:

A thread goes through various stages in its life for. eg. A thread is born, started, runs and then dies

Scanned with CamScanner

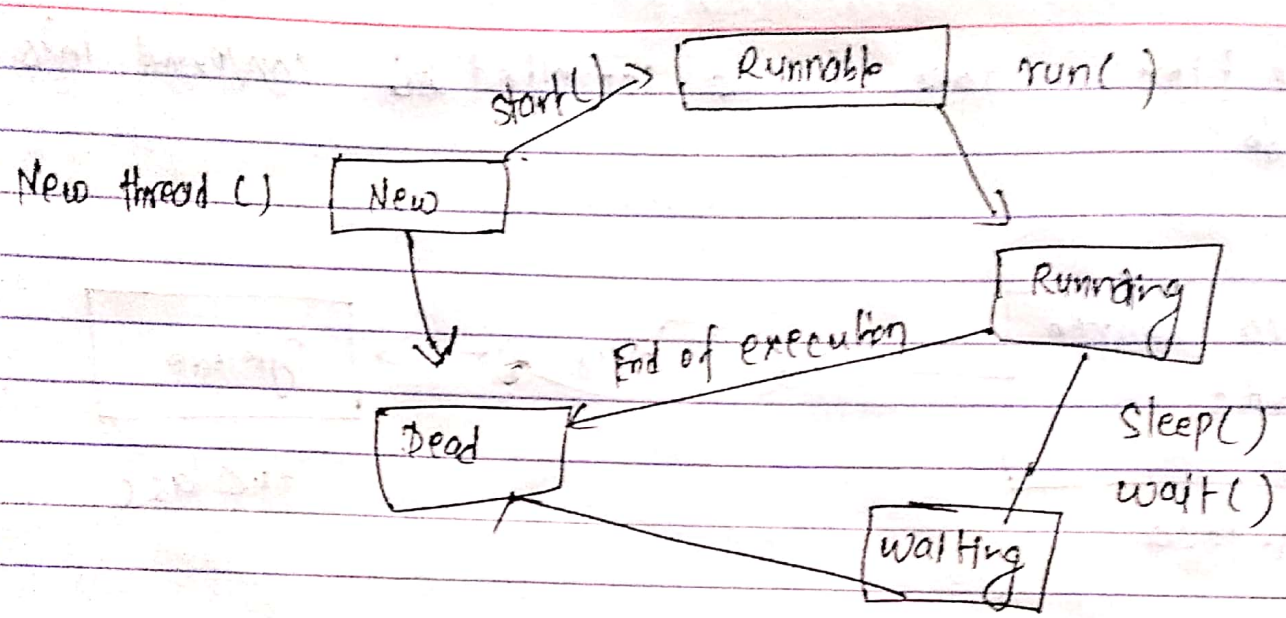


fig: life cycle of a thread

New : A thread has not yet started in this state

Runnable : A thread executing in the JVM is in this state.

Blocked : A thread that is blocked waiting for a monitor lock is in this state

Waiting : A thread that is waiting indefinitely for another thread to perform a particular action is in this state.

Terminated / Dead : A thread that has exited is in this state.

Scanned with CamScanner

Approaches for multithreading: Threads can be created by using two mechanisms

1) Extending the thread class

2) implementing the runnable interface

1) Thread creation by extending the thread class
We create a class that extends the Java lang thread class. This class overrides the run() method

example:

```
class A extends Thread {

  public void run () { try {
    System.out.println ("Thread" + thread.currentThread().getId() + " is running");
  }
  Catch (Exception e) {
    System.out.println("Exception is caught");
  }
}
}
```

Scanned with CamScanner

```
public class multithread {
public static void main( string args [ ])
{
    int n = 10;
    for ( int i = 0 ; i < n ; i++ )
    {
        A a = new A();
        a.start();
    }
}
}
```

2) · Thread Creation by implementing Runnable interface
We create a new class which implements java Runnable interface and Override run() method.

example:

```
class A implements Runnable {

    public void run () {
        try { System.out.println (" Thread" + thread.currentThread()
        getId () + " is running");
        }
```

Scanned with CamScanner

```
Catch ( Exception e )
{
System.out.println ("exception is caught").
    }
    }
    }
public class multithread {
public static void main (String args.[ ])
{
    int n = 10;
    for (int i = 0; i < n; i++)
    {
    Thread t = new thread (new A ());
      t.start ();
    }
    }
}
```

Scanned with CamScanner

P.N.3 What is Interface? write different properties of an interface with examples.

⇒ An Interface in Java is a blueprint of a class. It has static constants and abstract methods. It is used to achieve abstraction and multiple inheritance in Java. Java interface also represents IS-A relationship.

Example of declaring an Interface in Java

```
Public interface Car {
  public void start();
  public void run();
  public void stop();
}
```

Different properties of an Interface

1) Implementing Interfaces:

⇒ A class uses the implements keyword to implement an interface

Scanned with CamScanner

e.g

```
Public Class vechile implement Car {

Public void start()
{
    System.out.println ("car runs");
}
Public void run()
{
    System.out.println ("car is running");
}
```

4) What is Swing? write different Swing components. Write a program which gives the concept of container and components.

→ Java swing is used to create window-based application. It is built on the top of AWT. API and entirely written in Java. unlike AWT Java Swing provides Platform independent and light weight Components

Different Swing components are

1) Image Icon 2) JButton 3) JLabel
4) JPasswordfield 5) JTextfield 6) JTextArea

Scanned with CamScanner

7) JCheckbox 8) JRadioButton 9) JList
10) JComboBox 11) JFile chooser 12) JTabbedPane
13) JSlider

Program that shows container and components

```
import javax.swing*

public class A {

JFrame f;

A() {
  F = new JFrame();

  JButton b = new JButton("clicked");

  b.setBounds(130, 100, 100, 40);

  F.add(b);

  F.setSize(400, 500);

  F.setLayout(null);

  F.setVisible(true);
  } public static void main(String[] args) {
      new A();
```

Scanned with CamScanner

Q.N.5) what are different oop features in Java? write an object oriented program to find the area and circumference of a circle

→ The different oop features in Java are listed below

1) classes: A real world entity which a acts as a blueprint of all the objects

2) objects: object contains both variables (used for holding the data) and methods (used for defining the behaviours)

3) Abstraction: Abstraction is a process where you only shows only relevant data and hide unnecessary details of an an object from the user

4) Encapsulation: process of binding object state and behaviour together in a single entity called encapsulation

5) Inheritance: A process of defining a new class based on a existing class by extending its common data and members.

6) Polymorphism: It is the concept where an object behaves differently in different situations.

Scanned with CamScanner

Program to find area and Circumference of a Circle

Circle.java

```
public class Circle {
private float radius;
Private float getradius () {
    return radius;
    }
Public void setradius ( float radius) {
    this. radius = radius;
    }
Public static void main ( string [] args) {
    Circle c = new Circle();
    c.setRadius(5);
    Circle operation op = new operationImp();
    float area = op. finiArea(c.getRadius())
```

Scanned with CamScanner

```
float Circumference = op.findCircumference(c.getRadius());
System.out.println("Area is " + area);
System.out.println("Circumference is " + Circumference);
}
}
```

getter and Setter are used to achieve encapsulation.

Q.N.6) What are different File IO classes? write program for Byte stream and character Stream Classes.

⇒ Different File IO classes in Java are:

a) Byte stream class: Byte streams process data byte to Byte (8bits), for example FileInputStream to write to the destination

<u>program</u>

```
import java.io.*
```

Scanned with CamScanner

```
Public class copyfile{
Public static void main (String [] args)
throws IOException {
FileInputStream in = null;
FileOutputstream out = null;
try {
in = new FileInputstream ("input.text")
out = new FileInputstream ("out.txt");
int c;
while ((c = in.read()) != -1)
{
out.write(c);
}
}
finally {
```

Scanned with CamScanner

```
if (in != null)
{
  in.close();
}
if (out != null)
{ out.close(); }

}
}

}

}
```

Scanned with CamScanner

b) Character Stream class: In Java character are stored using unicode conventions. character stream automatically allows us to read/write data character by character. for example file reader and file writer are character streams used to read from source and write to destination

program

```
import java.io.*;

public class CopyFile{
Public static void main (String[] args) throws
IOexception{

    FileReader in = null;
    FileWriter out = null;

    try{
        in = new FileReader ("input.txt");
        out = new FileWriter ("output.txt");

        int c;

        while ((c = in.read()) != -1){

            out.write(c);
        }
    }
```

Scanned with CamScanner

ample: The following is an array declared with 2 elements then the code tries to access the third element of the array which throws an exception.

Custom exception in Java:

Java provides us facilities to create our own exceptions which are basically derived classes of exception. for example ExceptionTest class extends the Exception class. In example program we pass the string to the constructor of the super class Exception which is obtained using getmessage () method.

example

```
import Java.lo*;

public class ExceptionTest extends Exception {

public ExceptionTest (string s) {

    Super(s);

  }

}
```

Scanned with CamScanner

```
public class main{
public static void main (String [] args)
{
    try
    {
    throw new Exception test("caught");
    }
    Catch ( Exception test (e) {
    System.out.println("Exception caught");
    System.out.println (e.getmessage ()).
    }
    }
    }
```

Output:

Exception caught
Caught

Scanned with CamScanner

```
finally {
if (in != null)
  { ~~in.close~~
    in.close();
  }
  if (out != null)
  { out.close();
  }
  }
  }
  }
```

Scanned with CamScanner

Q.N.7) What are checked and unchecked exceptions? Explain the importance of exception handling with suitable example. What is custom exceptions, elaborate with an example.

⇒ An exception is a problem that arises during the execution of a program.

a) Checked exceptions: A checked exceptions is an exception that is checked by the compiler at compile-time, these are also called as a compile time exception

b) Unchecked exception: An unchecked exception is an exception that occurs at the time of execution, also called runtime exceptions.

* The importance of exception handling are

1) To continue program execution after an exception is caught and handled. The execution of a program does not terminate when an exception occurs.

try, catch and finally blocks are used in Java for exception handling.

Scanned with CamScanner

Q.N.8) Differentiate between Gridlayout and Gridbaglayout with examples.

⇒ A gridlayout puts all the components rectangular grid and its divide into sized rectangles. and each component inside a rectangle whereas Grid Baglayout a flexible layout manager that aligns components vertically and horizontaly without requiring that the components be of the same size.

Grid layout example

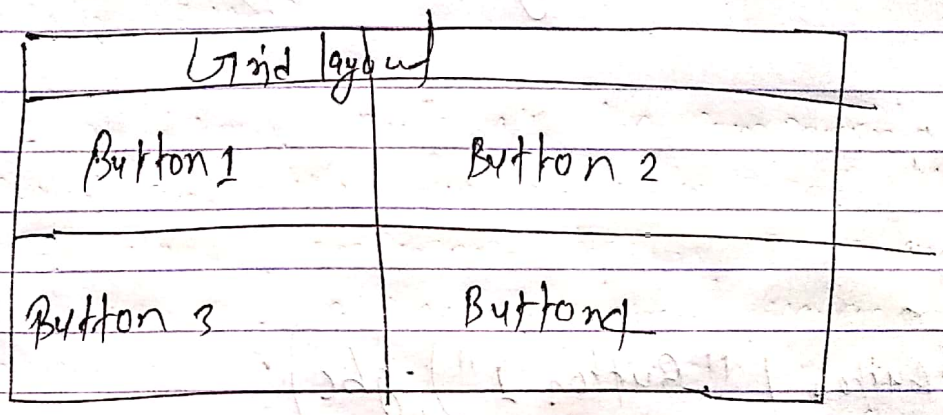
```
import java.awt.*;
import javax.swing.*;

public class GridLayoutTest {

GridLayout Test () {

JFrame frame = new JFrame ("Grid layout");

JButton btn1, btn2, btn3, btn4;

btn1 = new JButton ("Button1");

btn2 = new JButton ("Button2");
btn3 = new JButton ("Button3");
```

Scanned with CamScanner

```
btn4 = new JButton("Button 4");
frame.add(btn1)
frame.add(btn2)
frame.add(btn3)
frame.add(btn4)
frame.setlayout(new Gridlayout(2,2));
frame.setsize(300,300)
frame.setvisible(true);
}

Public static void main(String[] args)
{
    new Gridlayout test();
}
```

Scanned with CamScanner

Output



GridBaglayout example

```
import java.awt.*;
import javax.swing.*;
public class GridBaglayoutTest extends JFrame
Public GridBagLayoutTest() {
    setTitle("GridBag Layout");
    setLayout(new GridBagLayout());
```

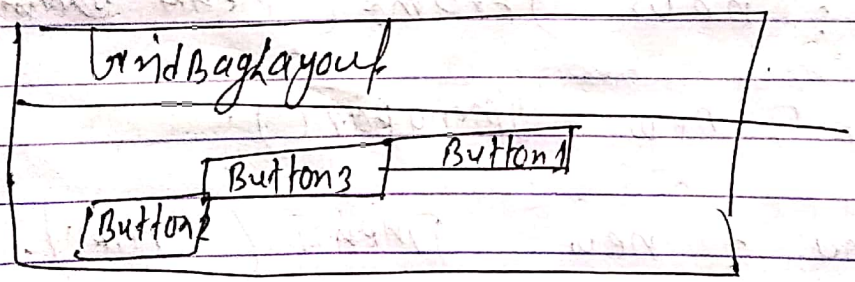
Scanned with CamScanner

```
GridBagLayout layout = new GridBagLayout();
gbc = new GridBagConstraints();

gbc.gridx = 5;
gbc.gridy = 0;
add(new JButton("Button 1"), gbc);
gbc.gridx = 0;
gbc.gridy = 5;
add(new JButton("Button 2"), gbc);
gbc.gridx = 2;
gbc.gridy = 4;
add(new JButton("Button 3"), gbc);
}
```

Scanned with CamScanner

```
public static void main (string[] args)
{
GridBagLayoutTest gbcTest = new GridBagLayoutTest();
gbcTest.setSize(300,300);
gbcTest.setVisible(true);
}
}
```

Output



Q.No.9) What are menu components, mnemonics, Accelerations and write a program to illustrate them?

→ Menurelated components are menu bar, menu items, radio button menu items, Checkbox, menu items and separators

Scanned with CamScanner

Mnemonics defines a key that lets us select an item from an active menu by typing the shortcut key

An accelerator is the key that lets us select a menu item without having to first activate the menu.

example.

```
import java.awt.*;
import javax.swing.*;

public class static void main(String [] ags)
{
    JFrame frame = new JFrame("Menu example");

    JMenuBar bar = new JMenuBar();

    JMenu fileMenu = new JMenu("File");

    fileMenu.setMnemonic(KeyEvent.VK_F);

    bar.add(fileMenu);
```

Scanned with CamScanner

```
JMenuItem mitem = new JMenuItem ("New
                keyEvent.VK_N);

FileMenu.add (mitem);

JMenuItem CutItem = new JMenuItem ("Cut",
                KeyEvent.VK_T)

KeyStroke ctr (KeyStroke = KeyStroke.getKeyStroke
                ("control X");

CutItem.SetAccelerator (ctr (KeyStroke));

FileMenu.add (CutItem);

Frame.SetJMenuBar (bar);

Frame.SetSize (300, 300)

Frame.SetVisible (true);

}

}
```

Scanned with CamScanner

Q.N. 10) What are packages and relate it with access Modifier? Illustrate with examples

⇒ Packages are Container for Classes that are used to keep the class name space

For. eg a package allows us to create a class named list which we can store in our own package without concern that it will collide with some other Class named list stored elsewhere Packages are stored in a hierarchial manner and its explicitly imported into a new class defination

Defining a package

package. package_name;

Access modifier

The three access modifier, private, public and protected, provide a variety of ways to produce the many levels of access required by those categories. table below sums up the interactions

Scanned with CamScanner

	Private	No modifier	Protected	Public
Same class	yes	yes	yes	yes
Same package subclass	No	yes	yes	yes
Same package Non subclass	No	yes	yes	yes
Different package subclass	No	No	yes	yes
Different package non subclass	No	No	No	yes

Q.N. 11) What are component organizers? write a program illustrating JDesktopPane and JInternalFrame

A) A component is a independent visual control. Java Swing framework contains a large set of these components which provide rich functionalities and allow high level of customization.

The JDesktopPane is a container which is used to create a multiple-document interface or a virtual desktop. The JFrame inside the desktop becomes JInternalFrame.

Scanned with CamScanner

* Programming illustrating JDesktopPane and JInternalfram

```
import javax.swing*;
import java.awt*
public class Test extends JFrame{
private JDesktopPane jd = new JDesktopPane()
Test()
{
setInternalFrame(jd, "Frame", 30, 30);
setInternalFrame(jd, "Frame2", 60, 60);
setDesktopPane();
setSize(700, 400);
setVisible(true);
}
```

Scanned with CamScanner

Q.N.12) What are advanced Swing components? Write a Java program that implements these components

⇒ Advanced swing components that helps us to deliver swing application with professional and friendly interface. Advanced swing components fit all popular look and feels the current version of AC suite contains Animated components such as JSlidepane, JTreetable Component, JSportable, Sortable and Filterable, JTable, Quick Search components Auto complete components, Advanced Toolbar and menubar, JSplashscreen, font components File components and others.

Q.N.13) What is event handling? What is a delegation model? Write three different types using delegation model

⇒ Change in the state of an object is known as event. Event handling is the mechanism that controls the event and decides what should happen if an event occurs this mechanism has a code which is known as an event handler, that is executed when an event occurs.

Scanned with CamScanner

Java uses the delegation event model handle the events. This model defines standard mechanism to generate and events.

The deligation event model has following participants

⇒ Source: the source is an object on which the event occur. Source is responsible for providing information of the occured event its handler.

⇒ Listner: It is also known as event handler the listner is responsible for generating a responsible to an event. the listner waits fill it receiver an event.

Q.N. 14) What is the difference between window listner and WindoAdapter? What is the difference between mouseAdapter and mouselistner? Describe with example

⇒ The class which process the window event should implement windowlistner interface the object of that class must be registered with a component. The object can be registered using the addwindowListner() method

Scanned with CamScanner

```
if (conn! = null)
    System.out.println("connected");
    }
    }
catch (SQLException ex)
    {ex.printStackTrace();
    }
```

We use con.close() to close connection explicitly after finishing working with database

* Inserting Data

```
String sql = "Insert into user (username, password, email) values('ram', 'ram12', 'hpegmail')
PreparedStatement ps = con.PreparedStatement(sql);
ps.setString(1, "Ram"
```

Scanned with CamScanner

```
ps.setString(2, "password");
ps.setString(3, "gopal@gmail.com");

int rowsInserted = ps.executeUpdate();

if (rowInserted > 0){
    System.out.println("Inserted successfully");
}
```

* update data

```
String sql = "update user set password=?
    email=? where username=?";

PreparedStatement ps = con.preparedStatement(sql);
    ps.setString(1, "password")
    ps.setString(2, "ram@gmail.com");
    ps.setString(3, "Ram");
```

Scanned with CamScanner

1.15) What is JDBC. show the implementation of Crud operation using JDBC.

) Java Database Connectivity is an application programming interface (API), which defines how a client may access a database. It is a Java base data access technolog used for Java data base Connectivity.

Crud operation using JDBC:

Step1

Creating user table in mysql database with user id, username, password and email where user id is autoincrement.

step2 : Connecting to mysql database on default port 3306

```
String dburl = "jdbc:mysql://localhost:3306/db";
String username = "root"
String password = "root"
try
    Connection con = DriverManager.getConnection
                    (db.URL, username, password);
```

Scanned with CamScanner

```
int rowsupdated = ps.executeupdate();
if (rowsupdated > 0)
{ System.out.println("update successfully");
```

Deleting data

```
String sql = "delete from user where username=?";
preparedstatement ps = con.prepareStatement(sql);
ps.setString(1, "Ram");

int RowDeleted = ps.executeupdate();
if (rowsDeleted > 0)
{ System.out.println("Deleted Successfully");
```

Scanned with CamScanner

* selecting data

```
String sql = "select * from user";
statement stat = con.Createstatement()
Resultset rs = stat.executequery(sql);
  int count = 0;
  while (rs.next())
  {   string name = rs.getstring(2);
      String pass = rs.getstring(3);
      String email = rs.getstring(4);
        System.out.println("name:" + name, "password":
        + pass +", " + "email:" + email);
  }
```

Scanned with CamScanner

Q.N.16) What is PreparedStatement? Show how it is better than generic statement with example

⇒ Once a Connection is obtained we can interact with the database, the PreparedStatement interface is a subinterface and PreparedStatement are used to execute static SQL query and parameterized SQL query respectively. Their commonly used methods are

boolean execute (String SQL) : execute a general SQL statement.

int executeUpdate (String SQL) - executes an Insert, update or Delete statements and returns on update account indicating number of rows affected.

ResultSet executeQuery (String SQL) - executes a select statement and returns a ResultSet object.

A prepared statement is one that contains placeholder (in form of question marks ?) for dynamic values will be set at runtime that's why PreparedStatement is better than generic statement if we want to set dynamic values

Example of inserting data using statement and PreparedStatement

Scanned with CamScanner

=) String firstrow = " insert into User (username, password) Values ("Ram", "pass")

String secondrow = " insert into user (username, password) Values ("shyam", "Password");

```
try {
    Statement stmt = con.Createstatement(firstrow);
    Statement start2 = Con.Createstatement(secondrow);
}
Catch block
```

Q.N. 27) What is Resultset and implement the properties of Resultset with example

=) Resultset contains table data returned by a select query, this object is used to iterate over rows in the result set using next() method and get value of a column in the current row using getInt(), getString() and so on methods. The column value can be retrieved either by column index number or by column name.

Scanned with CamScanner

Result navigational methods with example

→ boolean absolute (int row) - moves Resultset cursor to the specified row

→ boolean first() - moves Resultset cursor to the first row

→ boolean last() - Moves Resultset cursor to the last row

→ boolean next() - Moves Resultset cursor to the next row

→ boolean previous() - moves Resultset cursor to the next row

→ void afterlast() - moves resultset cursor to the position after last row

→ void beforefirst() - moves Resultset cursor to the position before the first row.

Examples of Resultset properties

Suppose we have usertable with three data

Scanned with CamScanner

Username : Ram
Password : Pass
Username : Shyam
Password : Password
Username = nppyalcurel
Password = np1234j

Program

```
String sql = select * from user
try {
Statement Stmt = Con.Createstatement();
Resultset rs = stmt.executeQuery("sql");
rs.absolute(2); PrintData(rs);
rs.previous();
PrintData(rs);
rs.last();
PrintData(rs);
```

Scanned with CamScanner

```
while (rs.next())
{
    PrintData(rs);
    }
    }
catch block

public static void printData (ResultSet rs)
throws SqlException
    System.out.println ("username", + rs.getstring(1))
    System.out.println ("password" + rs.getstring(2));
    }
```

Q.N.28) What is Transaction and what methods are used to implement the Transaction in Java?

⇒ A Transaction is a set of one or more statements that is executed as a unit, so either all of the statements are executed or none of statement is executed. The way to allow two or more statements to be grouped into a transaction is to disable the auto commit mode

Methods that are used to implement transaction in Java are

- void setAutoCommit (boolean status) → It is true by default which means each transaction is committed by default.

→ void commit () → commits the transaction

→ void rollback → Cancels the transaction

The following example illustrates the use of transaction properties

```
try {
    Con. SetAutoCommit( false);
    Statement start = con. CreateStatement ();
    String sql = " Insert onto user ( username, password)
      values ( "Ram", "Pass ");
      start. executeupdate (sql);
    String sql = " Insert in User ( username , password)
    values ( "sham", "pass");
      start. executeupdate(sql);
```

Scanned with CamScanner

```
con. Commit();
}
Catch (SQLException ex)
{
  con. Rollback();
}
```

In this case, none of the insert statement could succeed and every thing could be rolled back.

Q.N. 19) What is network programming? write about TCP and UDP with examples.

→ The term network programming refers to writing programs that executes across multiple devices. on which the devices are all connection to each other using a network. The Java net Package provides support for the two common network protocols.

→ TCP : Transmission control protocol allows reliable communication between two applications TCP is typically used over the internet protocol which is referred to as TCP/IP.

Scanned with CamScanner

UDP: UDP stands for User Datagram protocol a connectionless protocol that allows for packets of data to be transmitted between applications.

At the core of Java's networking support are the Socket and DatagramSocket class in java.net that select socket class is used for creating TCP connection over and IP network. A socket is typically creating using an InetAddress to specify the remote host and a port numbers to which the host has to connect.

examples of TCP on the client side

```
InetAddress addr = InetAddress.getByName("our.remote.host");
Socket s = new Socket(addr, 15000);
Input stream in = s.getInputStream();
```

UDP socket connection are created and used through the DatagramSocket and DatagramPacket classes. A Datagrampacket sends and receives data using UDP packets represented as Datagrampacket objects.

Scanned with CamScanner

DatagramSocket udpsocket = new DatagramSocket(5000)

If we dont care which port is used, then we can construct the DatagramSocket without specifying the port.

DatagramSocket udpsocket = new DatagramSocket();

int port-num = ~~updat~~ udpsocket.getlocalPort();

Q.N.10) What are jars in java? write steps of creating, reading, updating java file using command mode?

⇒ A Jar (Java Archive) is a package file format typically used to aggregate many java class files and associated metadata and resources into one file to distribute application software or libraries on the Java platform.

1) Creating a Jar file: To create a jar file we can use jar cf command.

→ Jar cf Jarfilename inputfiles
Here, cf represents create new file.

Scanned with CamScanner

) Viewing/Reading a jar file: To view the contents of jar we can use a command as;

→ Jar tf . Jarfilename
Here tf represents table view of file contents

3) updating Jar file: The Jar tools provides option which is used to update the contents of an existing Jar file.

4) Running a jar file: jar pack. Jar(Jar Jarfilename)

Q.N.21) What are Java beans? How Java beans can be created and executed? give examples.

→ A Java bean is a software component that has been designed to be reusable in a variety of different environment. There is no restriction on the capability of a bean. Beans are important because they allow to build complex system from software components.

Steps for creating a new bean

1) Create a directory of the new bean
2) Create a Java source file(s)
3) Compile the source file(s)
4) Create a manifest file

Scanned with CamScanner

5) Create a Jar file
6) Start the BDK (Bean development kit)
7) Test

Simple example of JavaBeans class

Employee.Java

```
public class Employee implements Java.io.Serializable
{
  Private int id;
  private string name;
  public Employee()
      {
      }
  Getter and setter

}
```

Accessing the bean class

```
Public class Test {
```

Scanned with CamScanner

```
public static void main( String [] args)
{
Employee e = new Employee();
e.setName("Lari");
System.out.println( e.getName() );
}
```

3
properties of Java bean class are:

1) . get.propertyName () . e.g - getName();

2) Set.propertyName () . e.g setName();

P.N. 22) What are Servlets? write a program to submit the data from html using post method

=> Servlets provide a Component based Platform independent method for building web-based applications. Servlets are small programs that execute on the server side of a web application using servlets we can collect input from users through web form, present record from a database or another source. and Create

Scanned with CamScanner

dynamic webpage

Example of post method using html and servlet.

html

```
<form name = "loginform" method = "post"
  action = "loginservlet" >
  Username = <input type = "text" name = "uname" >
  <br>
  password : <input type = "password" name = "pass" >
  <br>
  <input type = "submit" value = "login" >

  </form>
```

output

Username : [ ]

Password [ ]

[login]

Scanned with CamScanner

Q.N.23) How all parameters data can be retrieved in select " Define with example

→ Program to retrieve all parameters data in servlet

Simple html form

```
<form action = " DifferentForm Data" method = "post">

Name : <input type = "text" name = "fname"> <br>

Gender : <input type = "radio" checked name = "gender"
  value = "male"> Male <input type = "radio"
  checked name = "female"> Female <br>
Select Country : <select name = "country">

  <option value = "Nepal"> Nepal </option>

  <option value = "India"> India </option>

  </select> <br>

<input type = "Submit" value = "Submit">

</form>
```

Scanned with CamScanner

Different form Data Java

```
import javax.servelet*
import java.io*
import java.util.Enumeration

@webservelet

Public class DifferentformData extends HTTPservelet
{

Protected void doPost( HTTPServeletRequest req.
    HTTPServeletResponse res) throws ServeletException,

  {. IOExceptions
  res.setContentType("text/html");

  PrintWriter out = res.getWriter();

  Enumeration <String> Params = req.getparametersNames();
while(params.has.moreElements())
{
  String paramname = (String).params.nextelements
```

Scanned with CamScanner

```
out.println("<ul>");
out.print.ln("<li>");
out.println(request.getParameter(paramname));
out.println("</li>");
out.println("</ul>");
}
}
}
```

Q.N.24) What is the difference between Servlets and JSP? Write a program which prints "CORONA PANDIMIC" 20 times using JSP.

⇒ Difference between servlets and JSP.

Scanned with CamScanner

Servlet	Jsp
1) Servlet is developed on Java Progr language	1) Jsp is a Html based code
2) Servlet can accept all protocol requests	2) Jsp only accepts Http requests.
3) Servlet is faster than Jsp	3) Jsp is slower because Jsp is converted to Java code then compiled
4) In servlet, we implement business and presentation logic in just one servlet the service() method	4) In Jsp, business logic is separated from presentation logic by using Javabeans.
5) We can override the service() method.	5) We cannot override service() method.

Scanned with CamScanner

Program that prints "Corona pandemic" 20 times

```
<html>
<head>
<body>
<%for (int i=1; i<=20; i++)
{%}
<h5> Corona Pandemic </h5>
<br>
<%}%>
</body>
</html>
```

Q.N.25) Write a program to perform different database related statements in JSP.

⇒ Database related statements in JSP

Scanned with CamScanner

1) inserting data in mysal database through JSP

```
<%@ page import = "java.sql*"%>

<%@ page import = "java.io*"%>

<html>
 <head>
 </head>

 <body>
<form action = "form.jsp" method = "get">

Name : <input type : "text" name : "name">

Password : <input type = "password" name = "password">

    <input type : "submit" value = "submit".

    <%
    String name = request.getparameter ("name").
    string password = request.getparameter ("password")
    string conURL = "jdbc:mysal://localhost:33%/db;

    Connection con = null;
```

Scanned with CamScanner

```
preparedstatement ps = null;

Class forNam ("com.mysql.jdbc.Driver")

NewInstance ();
int updateQuery = 0;
try {
  Con = Drivermanager.getConnection (conURL,
        "root", "root");

  String query = "insert into User(name, password) value
        (?,?)";
  ps = Con.preparedstatement (query);
  ps = setstring(1, name);
  ps.setstring (2, password);
  updateQuery = ps.executeupdate();
  if (updateQuery != 0 { %>
  <h5> Data is inserted /h5>
  <%
  }

}
```

Scanned with CamScanner

```
catch ( Exception ex)
{
out.println ("unable to insert");
}
finally {
    ps.close();
    con.close();
  }
}
%>
</form></body></html>
```

Q.N.26) Difference between RMI and CORBA?

1) RMI is a Java-specific technology. CORBA has implementations for many languages. CORBA can be used to share objects between programs written in different languages

2) CORBA uses IDL (Interface Definition language) to seperate Interface from implementation. RMI just uses Java Interfaces

3) RMI programs can download new from remote JVMs. CORBA doesn't have this code sharing mechanism

4) CORBA is a more mature standard than RMI and has had time to gain richer implementations.

Scanned with CamScanner