RASHTREEYA SIKSHANA SAMITHI TRUST

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated To VTU, BELAGAVI)

**R.V. Vidyaniketan Post, Mysuru Road,
BENGALURU - 560 059**

MASTER OF COMPUTER APPLICATIONS

Lab Manual for Second Semester

JAVA BASED SOFTWARE SOLUTIONS

20MCA262

Faculty In-charge

Mr. Prashanth K

USN	
NAME	
Academic Year:	**2021-2022**

RASHTREEYA SIKSHANA SAMITHL TRUST (RSST)

The last decade of pre-independence India was marked with several initiatives and entrepreneurial ventures. One such unique venture was the founding of Rashtreeya Sikshana Samithi Trust (RSST), by Sri. M.C. Sivananda Sarma, a freedom fighter and a scholar in the year 1940. The organization started with a noble mission to impart quality education to all sections of the society, without any favour or bias towards anyone. Today, the charitable trust provides avenue for quality education, catering to a wide sector of educational needs, starting from kindergarten to post graduate education as well as research in advanced engineering, medical and architecture domains. The sustained growth and success is due to the dedicated efforts of the management and their continued commitment to the founder's vision, mission, quality, continuous improvement and concern towards social responsibility. Today the trust is managed by a very distinguished board of trustees led by **Dr. M.K. Panduranga Setty, President & Chairman Governing body, Sri. C.V. Hayagriv and Shri Panditharadhya, Vice Presidents, Shri K.G. Subbarama Setty, Hon. Treasurer, Shri A.V.S. Murthy, Hon. Secretary and Shri D.P. Nagaraj, Hon. Joint Secretary**. The board of trustees, recognizes the importance of holistic education and a need for a learning environment that nurtures healthy competition and innovation. Rashtreeya Sikshana Samithi Trust **manages over twenty educational institutions including schools, colleges offering degree, post graduate programs and doctoral programs in different specialties**. RSST continuously strives to create state of art infrastructure, recruit excellent faculty and facilitate efficient administration in all its institutions to provide congenial ambience for learning. Today, RSST is regarded as one of the finest and the best managements for education in the country.

RV COLLEGE OF ENGINEERING®

Marching towards Excellence in Education, Research and Innovation

Rashtreeya Vidyalaya College of Engineering (RVCE) established in 1963 is one of the earliest self-financing engineering colleges in the country. RV College of Engineering is the flagship institution of RSST. The institution provides opportunities to all sections, including the under privileged, differently abled and socially marginalized people to gain engineering skills through various programs like WEST-Women Empowerment and Skill Training etc. RVCE is rated amongst the top five self-financing Engineering colleges in the country. Some magazines have rated it as the best among private institutions in the country, including in terms of best Return on Investment for a student. RVCE is a preferred destination for top ranking aspirants, both for UG, PG and Doctoral programs. RVCE is an Autonomous college, affiliated to Visvesvaraya Technological University (VTU) Belagavi. The institution has its own Academic Council which is empowered to approve the academic curriculum as suggested by Board of Studies of various programs. RVCE currently offers 12 Bachelor, 21Master programs and 16 centers of research to carry out research and consultancy activities in the departments. All UG programs have been accredited multiple times. Some of the P.G. programs have been accredited and other eligible PG programs have applied for accreditation.

The Institution currently has student strength of about 5600, faculty strength of around 400 and around 300 Support Staff. The institution has set itself a Vision "Leadership in Technical Education, Interdisciplinary Research & Innovation, with a focus on Sustainable and Inclusive Technologies". All the departments are aligned to the vision of sustainable and inclusive technology development, with a focus to contribute to both technological leadership of the nation and welfare of all sections of the society. RVCE is rapidly expanding its R&D activity and Industry academic collaborations.

VISION

Leadership in Technical Education, Interdisciplinary Research & Innovation, with a Focus on Sustainable and Inclusive Technologies

MISSION

- *To deliver **O**utcome **B**ased **Q**uality **E**ducation, emphasizing on **e**xperiential learning with state-of-the-art infrastructure*
- *To **c**reate a conducive environment for **i**nterdisciplinary research and innovation*
- *To develop professionals through holistic education focusing on individual growth, discipline, integrity, ethics and social sensitivity*
- *To nurture industry-institution **c**ollaboration leading to competency enhancement and entrepreneurship.*
- *To focus on technologies that are **s**ustainable and **i**nclusive, benefiting all sections of the society*

PROFILE OF THE DEPARTMENT

The Department of Master of Computer Applications was established in year 1997 and is the first PG program started in RVCE. The programs offered by the department include, Masters of Computer Applications, M.Sc by Research and Ph.D. Degree. These programs are affiliated to Visvesvaraya Technological University, Belagavi. The program obtained academic autonomy in the year 2016. The sanctioned intake of students for first year of MCA is 120 students and additional 20% intake as lateral entry to 3rd Semester. The MCA program is accredited for second time by National Board of Accreditation, New Delhi since September 2013. Our graduates have the distinction of obtaining high positions in reputed IT industry. The faculties are from diverse background, committed, highly qualified with Doctorates in various specializations. They deliver quality education to students through their rich research experience. The faculties are engaged in active research works and projects funded by AICTE, NRB, DRDO agencies and industries to the tune of Rs.80.51 Lakhs have been completed in the last three years and ongoing research projects worth Rs. 29 Lakhs. Faculties have also taken up consultancy works and completed 3.25 Lakhs worth projects and have ongoing of 1 Lakh worth. The department has state-of-the-art infrastructure and computing facilities supported by high-speed Ethernet and wireless access

Vision

Pioneering in ICT Enabled Quality Education and Research with a focus on Sustainable and Inclusive Applications

Mission

- To adapt novel methodologies for quality education through experiential learning
- To empower students with continuous, holistic education, emphasizing on discipline, ethics and social commitment
- To become a vibrant knowledge center for research and software development
- To continuously build capacity steering towards industry- institute collaborative research and entrepreneurial competencies
- To utilize and develop free and open source software tools for sustainable and inclusive growth

Program Educational Objectives (PEO)

MCA graduates will be able to

PEO1: Practice software engineering principles and standards to develop software to meet customer requirements across verticals

PEO2: Contribute to build sustainable and inclusive applications using mathematical, simulation and meta-heuristic models

PEO3: Demonstrate entrepreneurial qualities through individual competence and teamwork.

PEO4: Achieve successful professional career with integrity and societal commitments leading to lifelong learning

Program Outcomes (PO)

MCA Graduates will be able to

PO1: Computational Knowledge: Acquire in-depth computational knowledge and mathematics with an ability to abstract and conceptualize models from defined problems and requirements

PO2: Problem Analysis: Identify, formulate, conduct literature survey and solve complex computing problems through analysis as well as provide optimal solutions

PO3: Design / Development of Solutions: Design and evaluate solutions for complex problems, components or processes that meet specified needs after considering public health and safety, cultural, societal and environmental factors

PO4: Conduct investigations of complex Computing problems: Conduct literature survey to analyze and extract information relevant to unfamiliar problems and synthesize information to provide valid conclusions and interpret data by applying appropriate research methods, tools and design experiments

PO5: Modern Tool Usage: Create, select, adapt and apply appropriate techniques, resources, and modern IT tools to complex computing system activities, with an understanding of the limitations

PO6: Professional Ethics: Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practices

PO7: Life-long Learning: Engage in lifelong learning independently for continual development to improve knowledge and competence as a computing professional

PO8: Project management and finance: Demonstrate knowledge and understanding of management principles and apply these to multidisciplinary software development as a team member and manage projects efficiently as a leader considering economical and financial factors

PO9: Communication Efficacy: Understand and communicate effectively with the computing community and with society at large, regarding complex computing systems activities confidently and effectively by writing effective reports and design documentations by adhering to appropriate standards, make effective presentations and give / receive clear instructions

PO10:Societal and Environmental Concern: Understand responsibilities and consequences based on societal, environmental, health, safety, legal and cultural issues within local and global contexts relevant to professional computing practices

PO11: Individual and Team Work: Function effectively as an individual, as a member or leader in diverse teams in multidisciplinary environments

PO12: Innovation and Entrepreneurship: Identify a timely opportunity for entrepreneurship and use innovation to pursue and create value addition for the betterment of the individual and society at large

Program Specific Criteria (PSC)

The MCA program will enable the students, by the time they graduate to:

PSC1: Explain the principles of mathematics, computing and business foundations
PSC2: Demonstrate the use of software tools and technologies relevant to various verticals
PSC3: Design and develop software products, processes and systems for real world situations

Program Specific Outcomes (PSO)

MCA graduates will be able to:

PSO1: Solve real world computing system problems of various industries by understanding and applying the principles of mathematics, computing techniques and business concepts
PSO2: Design, test, develop and maintain desktop, web, mobile and cross platform software applications using modern tools and technologies

RV COLLEGE OF ENGINEERING®, BENGALURU – 560059

(Autonomous Institution Affiliated to VTU, Belagavi)

Department of Master of Computer Applications

CERTIFICATE

This is to certify that Mr./Ms ……………………………………………………………

USN ……………………………..….. of 2nd **Semester** Master of Computer Applications program has satisfactorily completed the course of experiments in practical **JAVA BASED SOFTWARE SOLUTIONS – 20MCA262** prescribed for the academic year **2020 – 2021**.

LAB MARKS	
Max	**Obtained**
50	

Signature of Student **Signature of faculty In-Charge** **Signature of Director**

EVALUATION SHEET

Week	Prog. No.	Program Title	Date of Execution	Marks	Signature of Lab In-charge
1.	1.	Write a Java application for Education Domain which demonstrates the following scenario a. scope of variable b. this keyword			
2.	2.	Write a Java application for Education Domain which demonstrates the following scenario a. constructor overloading and method overloading b. static keyword			
3.	3.	Write a Java application to demonstrate the following concepts a. Inheritance and interface			
4.	4.	Write a Java application to demonstrate the following concepts a. packages			
5.	5.	Write a Java application to insert data into any DATABASE and retrieve info based on particular queries (For example update, delete, search etc...) with proper usage of user defined Exception Handling (also make use of throw, throws.) (Ex: if input exceeds the greater than the text field then user defined exception can be given)			
6.	6.	Write a Java application to insert data into any DATABASE with the usage of user defined Exception Handling (also make use of throw, throws.) (Ex: if input exceeds the greater than the text field then user defined exception can be given)			
7.	7.	Write a JAVA Servlet Program to Auto Web Page Refresh			
8.	8.	Write a JAVA Servlet Program to implement a dynamic HTML for following scenarios a. User name and password should be accepted in HTML b. Verify the username and password using a ServletConfig init-param and display the appropriate message on another Servlet			
9.	9.	Write a JSP program to demonstrate for a given scenario for Session API and cookies			
10.	10.	Write a JSP program to demonstrate for a given scenario for Conditional and exceptional handling techniques			
		Lab Manual Marks		**/100**	
		CIE Weekly Evaluation		**/40**	
		LAB Internals		**/10**	
		Total		**/50**	

Student Signature

INDEX

Laboratory Component

CIE Marks(Lab):	**50**	**SEE Marks(Lab):**	**50**
Hr/Week:	**2**	**SEE Hours:**	**03**

Prog. No.	**Program Title**
1.	Write a Java application for Education Domain which demonstrates the following scenario a. scope of variable b. this keyword c. constructor overloading and method overloading d. static keyword
2.	Write a Java application to demonstrate the following concepts a. Inheritance and interface b. packages
3.	Write a Java application to insert data into any DATABASE and retrieve info based on particular queries (For example update, delete, search etc...) with proper usage of user defined Exception Handling (also make use of throw, throws.) (Ex: if input exceeds the greater than the text field then user defined exception can be given)
4.	Write a Java Servlet Program to implement a dynamic HTML for the following scenario a. Display the auto web page refresh using HTTPRequest and HTTPResponse b. Verify the user authentication using ServletConfig
5.	Write a JSP program to demonstrate for a given scenario a. Session API and cookies b. Conditional and exceptional handling techniques

Note: Each program should be completed and executed within two consecutive lab cycles. However, for CIE test and SEE evaluation the examiner can specify minimum functionalities related to the programs keeping time factor as a constraint.

Scheme of Continuous Internal Evaluation (CIE) for Practical:

CIE for the practical will be based on the performance of the student in the laboratory every week for 10 marks for every experiment. Finally, the weekly evaluated marks will be consolidated for 40 marks. One test will be conducted at the end of the semester for 10 marks. The total marks for CIE (Practical) will be for 50 marks. No change of program is allowed.

Scheme of Semester End Evaluation (SEE) for Practical:

SEE for the practical will be based on writing proper program, execution and proper results and viva for 50 marks. No change of program is allowed.

Break-up Scheme for Continuous Internal Evaluation for Laboratory

Continuous Internal (Evaluation Laboratory)	Total Marks	Scale Down to
Performance of the student weekly in laboratory for every program (10 Marks)	For 10 Lab Programs 10 x 10 = 100	40
Internal Test (50 Marks) Write Up (20% of 50 Marks) Program Execution (60% of 50 Marks) Viva-Voce (20% of 50 Marks)	Total: 50 Marks	10
	Total CIE	50

CONTINUOUS INTERNAL EVALUATION					
RUBRICS FOR LAB PROGRAM CONDUCTION AND EXECUTION					
Sl. No	Criteria	Marks Allotted (Max: 7 marks)			Total Marks
		Excellent	Good	Poor	
1	Knowledge / Understanding of Concepts and Logic (2 marks)	Demonstrates the ability of describing concepts & logic (1.5- 2.0)	Demonstrates considerable ability in describing the concepts & logic (1.4 - 1.0)	Unable to demonstrate, describe& illustrate the concepts & logic (0)	7
2	Algorithmic Approach (2 marks)	Excellent / different Analysis of design and development to get expected output (1.5- 2.0)	Good Analysis of design and development to get expected output (1.4 - 1.0)	Poor Analysis of design & development to get expected output (0)	
3	Conduction and Execution of the program and Output (3 marks)	Conduct the program by inputting correct data and excellent presentation of the results (2-3)	Conduct the program by inputting correct data and fair presentation (1-1.9)	Conduct the program by inputting correct data and poor presentation of the results (<= 1)	
RUBRICS FOR VIVA-VOCE MARKS (MAX: 3 MARKS)					
4	Program Analysis (1 mark)	Answers to questions and demonstrates complete understanding of the working (1.0)	Answers to Questions &demonstrates good understanding of the working (0.5)	Answers to questions and demonstrates poor understanding of the working (0)	3
5	Application (1 mark)	Sound knowledge about applicability (1.0)	Demonstrates good knowledge about applicability (0.5)	Demonstrates very limited knowledge about applicability (0)	
6	Communication (1 mark)	Communicates all ideas clearly about method/ procedure (1.0)	Explains the procedure without clear understanding (0.5)	No idea about the procedure / method (0)	

RV COLLEGE OF ENGINEERING®, BENGALURU – 560059

(Autonomous Institution Affiliated to VTU, Belagavi)

Department of Master of Computer Applications

Java Based Software Solutions – 20MCA262

General Guidelines

- Use only vi editor
- Programs must be indented appropriately
- Students are required to explore the different commands and file system of UNIX
- Students are encouraged to try implementing the program requirements using different approach

Do's

- Come prepared to the Lab.
- Submit your Records to the staff In-charge and sign in the Log Book on entering the Lab.
- Follow the Lab exercise cycles as instructed by the Department. Violating the same will result in deduction of marks.
- Use the same login (if any) assigned to you.
- Put the chairs back to its position before you leave.
- Backlog exercises to be executed after completing regular exercises.

Don'ts

- Move around in the lab during the lab session.
- Tamper System Files or Try to access the Server.
- Write Data Sheets or Records in the Lab.
- Change the system assigned to you without the notice of the Lab Staff.
- Teaching your friends.

• *Prerequisites*

Students should have the basic knowledge of Object Oriented Programming.

Course Learning Objectives

Graduates shall be able to

1. Explain the concepts in Java programming Language
2. Demonstrate Java, Servlets and JSP for web applications
3. Apply Java concepts in developing Web based application
4. Analyze Java concepts in developing Servlets, JSP web application

Expected Course Outcomes

After going through this course, the student will be able to:

CO1: Understand the basic concepts of Java Application Programming
CO2: Identify and Apply various technologies for Java application programming
CO3: Demonstrate various problem solving methods for developing application
CO4: Analyze solutions using Java API concepts for real world applications

RV COLLEGE OF ENGINEERING® , BENGALURU – 560059

(Autonomous Institution, Affiliated to VTU, Belgavi)

JAVA BASED SOFTWARE SOLUTIONS – 20MCA262

INTRODUCTION

Java Based Software Solutions covers the basic topics of Java fundamentals, Servlets, Java Server Pages (JSP).

The JAVA Language:

The Origin of Java: Java was conceived by James Gosling and others at Sun Microsystems in 1991. This Language was initially called “Oak” but was renames “Java” in 1995.

Java's Contribution to the internet:

- **Java Applets :** And applet is a special kind of java program that is designed to be transmitted over the internet ant automaticaly executed by a Java-compatible web browser.
- **Security :** Java achieved this protection by confining an applet to the Java execution environment and not allowing it access to other parts of the computer.
- **Portability:** Portability is a major aspect of the Internet because there are many different types of computers and operating systems connected to it.
- **Java's Solution:** the Bytecode: Bytecode is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM).

Evolution of Java:

The Key Attributes of OOP

- **Encapsulation**: Encapsulation is the mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse.
- **Polymorphism**: Polymorphism (from Greek, meaning “many forms”) is a feature that allows one interface to be used for a general class of actions.
- **Inheritance**: Inheritance is the process by which one object acquires the properties of another object.

The Java Development Kit:

The JDK supplies two primary programs. The first is javac, which is the java compiler. It compiles your source code into bytecode. The second is java, Sometimes referred to as the application launcher. This is the program you will use to run a java program. It operates on the bytecode, using the JVM to execute your program.

A First Simple Program

```
/*
This is a simple Java program.
Call this file "Example.java".
*/
class Example {
// Your program begins with a call to main().
public static void main(String args[]) {
System.out.println("This is a simple Java program.");
}
}
```

First Sample Program Line by Line:

```
/*
This is a simple Java program.
Call this file "Example.java".
*/
```

This is a multiline comment.

The next line in the program is the single-line comment, shown here:

```
// Your program begins with a call to main().
```

The next line of code is shown here:

```
public static void main(String args[]) {
```

All Java applications begin execution by calling main( ). The full meaning of each part of this line cannot be given now, since it involves a detailed understanding of Java's approach to encapsulation. However, since most of the examples in the first part of this book will use this line of code, let's take a brief look at each part now.

The **public** keyword is an access specifier, which allows the programmer to control the visibility of class members. The keyword static allows main( ) to be called without having to instantiate a particular instance of the class. The main( ) is the method called when a Java application begins. Keep in mind that Java is case-sensitive. Thus, Main is different from main. It is important to understand that the Java compiler will compile classes that do not contain a main( ) method. In main( ), there is only one parameter, albeit a complicated one. String args[ ] declares a parameter named args, which is an array of instances of the class String. (Arrays are collections of similar objects.) Objects of type String store character strings. In this case, args receives any command-line arguments present when the program is executed.

The next line of code is shown here. Notice that it occurs inside main( ).

System.out.println("This is a simple Java program.");

This line outputs the string "This is a simple Java program." followed by a new line on the screen. Output is actually accomplished by the built-in println( ) method. Println( ) displays the string which is passed to it.

The first } in the program ends main( ), and the last } ends the Example class definition.

Handling Syntax Errors:

```
t.java:9: error: ';' expected
int a1=System.in.read()
^
1 error
```

the first error message is completely wrong because what is missing is not a semicolon but a brace.

JDBC

The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a Relational Database.

The following simple code fragment gives a simple example of these steps:

Steps for connecting to Database from Java program:

Step 1: Loading the driver

```
        Class.forName("com.mysql.cj.jdbc.Driver");
```

Your driver documentation will give you the class name to use. It is used to create an instance of a driver and register it with the DriverManager. When you have loaded a driver, it is available for making a connection with a DBMS.

Step 2: Creating a Connection:

Connection con = DriverManager.getConnection("jdbc:mysql://IPAddress/Database", "username", "password"**);**

Step 3: Creating a Statement

A Statement object is what sends your SQL statement to the DBMS. You simply create a Statement object and then execute it, supplying the appropriate execute method with the SQL statement you want to send. For a SELECT statement, the method to use is executeQuery. For statements that create or modify tables, the method to use is executeUpdate. It takes an instance of an active connection to create a Statement object. In the following example, we use our Connection object con to create the Statement object

Statement stmt = con.createStatement();

Step 4: Execute Query:

JDBC returns results in a ResultSet object, declare an instance of the class ResultSet to hold results. The following code demonstrates declaring the ResultSet object rs.

ResultSet rs = stmt.executeQuery(”Select Statement ”);

String s = rs.getString(”COF_NAME”);

The method getString is invoked on the ResultSet object rs, so getString() will retrieve (get) the value stored in the column in the current row of rs.

This short code fragment instantiates a DriverManager object to connect to a database driver and log into the database, instantiates a Statement object that carries your SQL language query to the database; instantiates a ResultSet object that retrieves the results of your query, and executes a simple while loop, which retrieves and displays those results. It's that simple.

Install a JDBC driver from the vendor of your database

If you are using Java DB, it already comes with a JDBC driver. If you are using MySQL, install the latest version of Connector/J.

Contact the vendor of your database to obtain a JDBC driver for your DBMS.

There are many possible implementations of JDBC drivers. These implementations are categorized as follows:

- **Type 1**: Drivers that implement the JDBC API as a mapping to another data access API, such as ODBC (Open Database Connectivity). Drivers of this type are generally dependent on a native library, which limits their portability. The JDBC-ODBC Bridge is an example of a Type 1 driver.

Note: The JDBC-ODBC Bridge should be considered a transitional solution. It is not supported by Oracle. Consider using this only if your DBMS does not offer a Java-only JDBC driver.

- **Type 2**: Drivers that are written partly in the Java programming language and partly in native code. These drivers use a native client library specific to the data source to which they connect. Again, because of the native code, their portability is limited. Oracle's OCI (Oracle Call Interface) client-side driver is an example of a Type 2 driver.
- **Type 3**: Drivers that use a pure Java client and communicate with a middleware server using a database-independent protocol. The middleware server then communicates the client's requests to the data source.
- **Type 4**: Drivers that are pure Java and implement the network protocol for a specific data source. The client connects directly to the data source.

This Trail uses the first two of these these four JDBC components to connect to a database and then build a java program that uses SQL commands to communicate with a test Relational Database. The last two components are used in specialized environments to test web applications, or to communicate with ODBC-aware DBMSs.JDBC Architecture

Two-tier and Three-tier Processing Models

The JDBC API supports both two-tier and three-tier processing models for database access.In the two-tier model, a Java applet or application talks directly to the data source. This requires a JDBC driver that can communicate with the particular data source being accessed. A user's commands are delivered to the database or other data source, and the results of those statements are sent back to the user. The data source may be located on another machine to which the user is

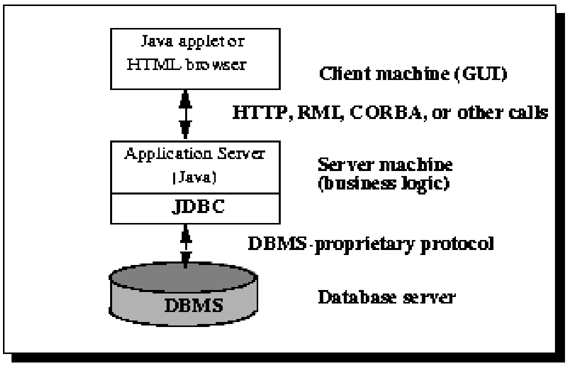

Java applet or HTML browser
Client machine (GUI)
HTTP, RMI, CORBA, or other calls
Application Server (Java)
JDBC
Server machine (business logic)
DBMS-proprietary protocol
DBMS
Database server

Figure 1: Two-tier Architecture for Data Access.

This is referred to as a client/server configuration, with the user's machine as the client, and the machine housing the data source as the server. The network can be an intranet, which, for example, connects employees within a corporation, or it can be the Internet.

In the three-tier model, commands are sent to a "middle tier" of services, which then sends the commands to the data source. The data source processes the commands and sends the results back to the middle tier, which then sends them to the user. MIS directors find the three-tier model very attractive because the middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data. Another advantage is that it simplifies the deployment of applications. Finally, in many cases, the three-tier architecture can provide performance advantages.

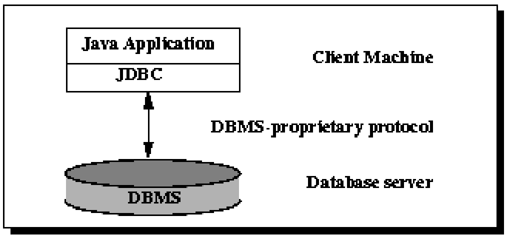


Figure 2: Three-tier Architecture for Data Access.

With enterprises increasingly using the Java programming language for writing server code, the JDBC API is being used more and more in the middle tier of a three-tier architecture. Some of the features that make JDBC a server technology are its support for connection pooling, distributed transactions, and disconnected rowsets. The JDBC API is also what allows access to a data source from a Java middle tier.

Java 2 Enterprise Edition:

J2EE is a platform-independent, Java-centric environment from Sun for developing, building and deploying Web-based enterprise applications online.

Java 2 Enterprise Edition Architecture:

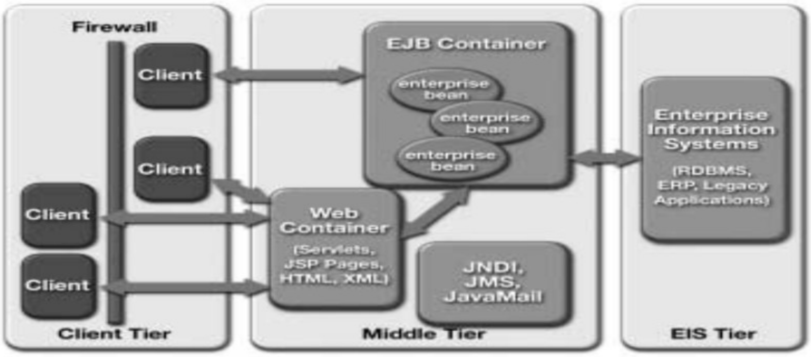


Figure 1.1 J2EE Environment

Apache tomcat server and its components:

Apache is an HTTP Server, serving HTTP. Tomcat is a Servlet and JSP Server serving Java technologies. Tomcat is a servlet container. A servlet, at the end, is a Java class. JSP files are generated into Java code (HttpServlet), which is then compiled to .class and executed by the Java virtual machine.

Tomcat 8.x was released with Catalina (a servlet container), Coyote (an HTTP connector) and Jasper (a JSP engine).

Catalina : Catalina is Tomcat's servlet container. Catalina implements Sun Microsystems' specifications for servlet and JavaServer Pages (JSP).

Coyote : Coyote is a Connector component for Tomcat that supports the HTTP 1.1 protocol as a web server. Coyote listens for incoming connections to the server on a specific TCP port and forwards the request to the Tomcat Engine to process the request and send back a response to the requesting client.

Jasper : Jasper is Tomcat's JSP Engine. Jasper parses JSP files to compile them into Java code as servlets (that can be handled by Catalina). At runtime, Jasper detects changes to JSP files and recompiles them.

Cluster : This component has been added to manage large applications. It is used for load balancing that can be achieved through many techniques. Clustering support currently requires the JDK version 1.5 or later.

High availability : A high-availability feature has been added to facilitate the scheduling of system upgrades (e.g. new releases, change requests) without affecting the live environment.

Web application : It has also added user- as well as system-based web applications enhancement to add support for deployment across the variety of environments.

Deployment Descriptor(web.xml):

Java web applications use a deployment descriptor file to determine how URLs map to servlets, which URLs require authentication, and other information. This file is named web.xml, and resides in the app's WAR under the WEB-INF/ directory. Web.xml is part of the servlet standard for web applications.

Web Components

A Web component is a software entity that provides a response to a request. A Web component typically generates the user interface for a Web-based application.

The J2EE platform specifies two types of Web components: servlets and JavaServer Pages (JSPTM) pages. The following sections give an overview of Web components.

1) Servlets:

A Servlet is a component that extends the functionality of a Web server in a portable and efficient manner. A Web server hosts Java Servlet classes that execute within a servlet container. The Web server maps a set of URLs to a Servlet so that HTTP requests to these URLs invoke the mapped Servlet. When a Servlet receives a request from a client, it generates a response, possibly by invoking business logic in enterprise beans or by querying a database directly. It then sends the response—as an HTML or XML document—to the requestor.

A servlet developer uses the Servlet API to:

- Initialize and finalize a servlet
- Access a Servlets environment
- Receive/forward requests and send responses
- Maintain session information on behalf of a client
- Interact with other Servlets and other components
- Use a filter mechanism for pre- and post-processing of requests and responses
- Implement and enforce security at the Web tier

2) JavaServer Pages:

The JavaServer Pages (JSP) technology provides an extensible way to generate dynamic content for a Web client. A JSP page is a text-based document that describes how to process a request to create a response. A JSP page contains:

- Template data to format the Web document. Typically the template data uses HTML or XML elements. Document designers can edit and work with these elements on the JSP page without affecting the dynamic content. This approach simplifies development because it separates presentation from dynamic content generation.
- JSP elements and scriptlets to generate the dynamic content in the Web document. JSP technology is extensible through the development of custom actions, or tags, which are encapsulated in tag libraries.

PROGRAMS

Program No. 1

Write a Java application for Education Domain which demonstrates the following scenario

a. scope of variable
b. this keyword
c. constructor overloading and method overloading
d. static keyword

PART 1.A

Part 1.A. includes the java application development in the education domain for

a. Scope of variables
b. This keyword

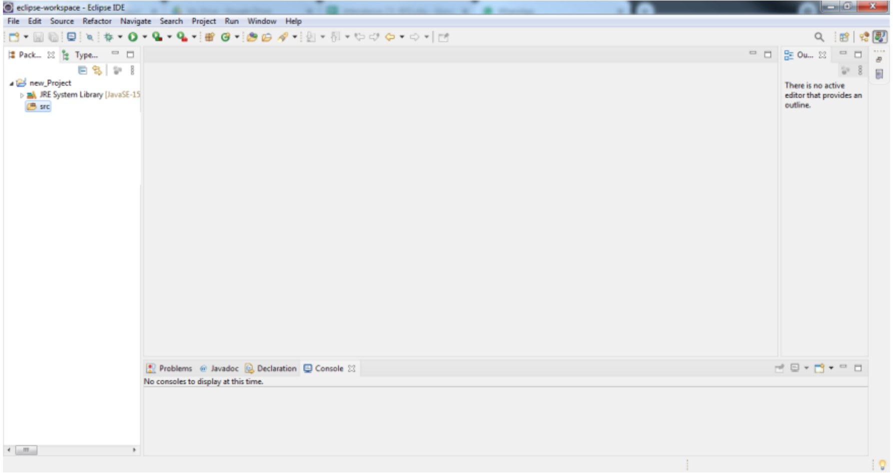
Question: write a java program to

a) Accept student information like (usn, name, address, emailid, branch)
b) Create a method to initialize the instance variables with student information
c) Fee calculation to be done in method based on the branch and display
d) Dynamically fill the details of class room
(Instance variables to be filled with details)
Note: use of this keyword to be demonstrated

Initial Steps for creating the project in Eclipse IDE: (these steps is same program1,2,3)

- **Step 1:** Create a Java Project in Eclipse IDE
  - Select - File → New → Java Project
- **Step 2:** Click on next button
- **Step 3:** Provide the suitable Project Name, click on finish button
  (Note: do not provide the java keywords as project name)
- **Step 4:** Project will be created as shown in the Eclipse IDE

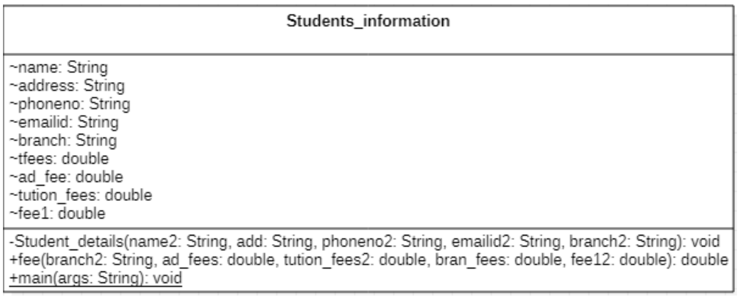
The figure shows the project tab at the left


eclipse-workspace - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Pack...
Type...
new_Project
JRE System Library [JavaSE-15
src
Ou...
Problems
Javadoc
Declaration
Console

There is no active editor that provides an outline.

No consoles to display at this time.

Fig: Shows the Eclipse IDE for new project creation



```
Students_information

~name: String
~address: String
~phoneno: String
~emailid: String
~branch: String
~tfees: double
~ad_fee: double
~tution_fees: double
~fee1: double

-Student_details(name2: String, add: String, phoneno2: String, emailid2: String, branch2: String): void
+fee(branch2: String, ad_fees: double, tution_fees2: double, bran_fees: double, fee12: double): double
+main(args: String): void
```

Fig: UML diagram – Class diagram

Step wise Execution Procedure:

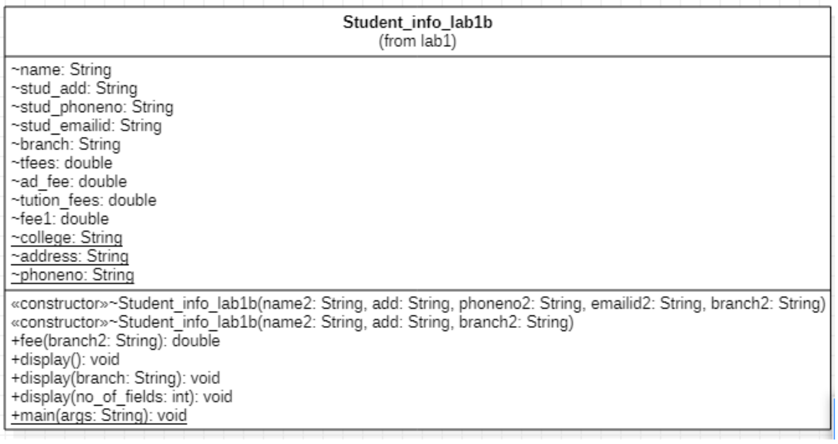
Students should demonstrate the **scope** of the **variables** and **this** keyword, by solving the above given question

- **Step 1:**
    - Students have to create a class with instance variables like usn, name, address, emailid, branch
- **Step 2:**
    - A constructor to be created to instantiate the instance variables from the accepted student information
- **Step 3:**
    - A method which should accept the fee details from the main program, calculate the total fees and return the value
- **Step 4:**
    - Execution to be shown with the proper format

Expected Output:

- The program should display the student information with proper formatting

PART 1.B



```
Student_info_lab1b
(from lab1)

~name: String
~stud_add: String
~stud_phoneno: String
~stud_emailid: String
~branch: String
~tfees: double
~ad_fee: double
~tution_fees: double
~fee1: double
~college: String
~address: String
~phoneno: String

«constructor»~Student_info_lab1b(name2: String, add: String, phoneno2: String, emailid2: String, branch2: String)
«constructor»~Student_info_lab1b(name2: String, add: String, branch2: String)
+fee(branch2: String): double
+display(): void
+display(branch: String): void
+display(no_of_fields: int): void
+main(args: String): void
```

Part B includes the java application development in the education domain for

a. Constructor overloading and method overloading
b. Static keyword

Question: write a java program to

a. Initialize the some of the instance variable using static block (Like college name, college address etc)
b. Accept the student information partial or full (Object should be created based on parameters created and by default rest of the values will be filled)
c. Display the details based on the student information (Method to be overloaded – display all the details of the student)

Step wise Execution Procedure:
Students should demonstrate the **constructor** and **method** overloading

- **Step 1:**
    - Students have to create a class with instance variables like usn, name, address, emailid, branch, college, address etc
- **Step 2:**
    - A static block has to initialize the details like college, college-address etc
- **Step 3:**
    - Two or three constructors to be created for student information
    - Constructors should accept and initialize the missing values with default values
    - Constructors should initialize the instance variables from the accepted student information
- **Step 4:**
    - Two or three overloaded display method to be created for displaying the student information back to the screen.
- **Step 5:**
    - Execution to be shown with the proper format

Expected Output:

- The program should create the objects with suitable constructors and display the student information with proper formatting

Sl.No	Rubrics for Practice Sessions		Max. Marks	Marks Obtained 1.a	Marks Obtained 1.b
1	**Conduction & Execution**	Understanding of Concepts and Logic	2		
2		Approach Towards the Problem	2		
3		Conduction and Execution: Input & Output for All Possible Cases	3		
4	**Viva Voce**	Program Analysis & Applications	2		
5		Communication & Confidence Level	1		
		TOTAL	10		
		Signature			

Program No. 1

Paste your DATA SHEET here

[Page intentionally left blank, student to paste his/her data sheet after evaluated by the staff in-charge]

Program No. 2

Write a Java application to demonstrate the following concepts

a. Inheritance and interface

b. Packages

PART 2.A

Part A includes the java application demonstration of Inheritance and interfaces

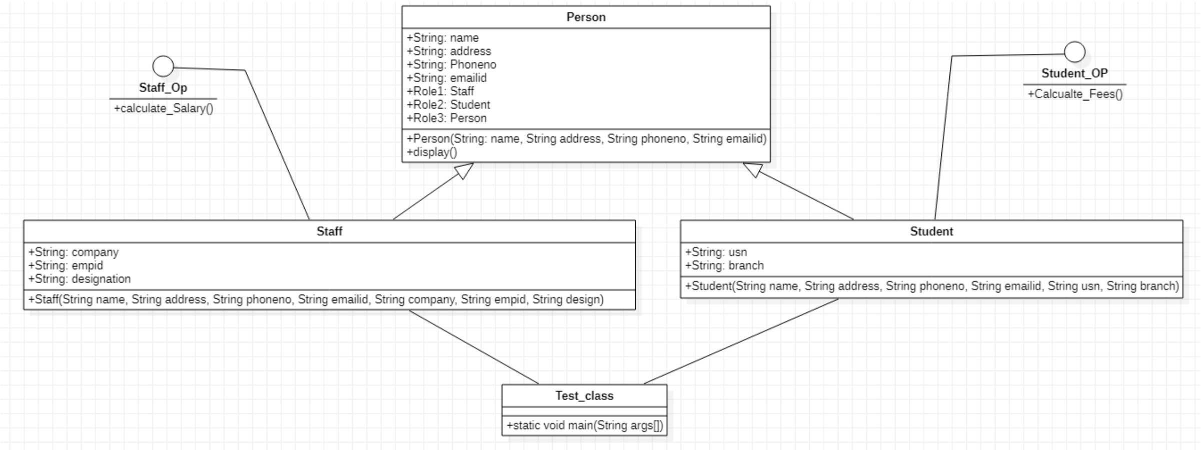

Person
+String: name
+String: address
+String: Phoneno
+String: emailid
+Role1: Staff
+Role2: Student
+Role3: Person
+Person(String: name, String address, String phoneno, String emailid)
+display()
Staff_Op
+calculate_Salary()
Student_OP
+Calcualte_Fees()
Staff
+String: company
+String: empid
+String: designation
+Staff(String name, String address, String phoneno, String emailid, String company, String empid, String design)
Student
+String: usn
+String: branch
+Student(String name, String address, String phoneno, String emailid, String usn, String branch)
Test_class
+static void main(String args[])

Fig: UML diagram – Class diagram

Question: write a java program:

a. Create person, employee and staff as classes

b. Staff and Student class should be defined with constructor

c. Staff and Student class to be extended with person.

d. Create interfaces for the class staff and student

e. Demonstrate with inheritance and interfaces using the main method in another class

Step wise Execution Procedure:

Students should demonstrate the **Inheritance** and **Interface** keyword, by solving the above given question

- **Step 1**:
    - Create a class with person, Staff and Student with respective instance variables like empid, company etc for each of the class for staff and
    - Create a constructor for Staff and Student respectively and initialize the person class parameters
- **Step 2**:
    - Create the interface Staff_Op and Student_Op for implementing the interfaces into the respective class.
- **Step 3**:
    - Create a class with main method and implement for the inheritance and interface
    - Object should be created and proper input of the data to be demonstrated for Student and Staff class

Expected Output:

- The program should display the both the Staff and Student with proper input

PART 2.B

Part B includes the java application demonstration of Packages

Question: Write a Java program to

a. Create a package named shape
b. Create classes in the package representing common shapes like Square, Triangle, and Circle
c. Import and compile these classes in other program

Step wise Execution Procedure:

Step 1: Creating the package with class.

1.1: In the project tab on source packages→ right click → new → Java Class. A wizard for the class will be popped up and in that give the class name as square. In the package field write Shapes in the same window. Repeat the step 1.1 three times for creating Triangle and Circle.

Step 2: Importing the package into the main class:

2.1: import the package Shapes

2.2: Create the objects of the classes

2.3: Call the respective methods of the classes.

Step 3: Right click on the source of the main class → click on Run File

Expected Output:

The program should display the methods called from the respective classes.

Sl.No	Rubrics for Practice Sessions		Max. Marks	Marks Obtained 2.a	Marks Obtained 2.b
1	**Conduction & Execution**	Understanding of Concepts and Logic	2		
2		Approach Towards the Problem	2		
3		Conduction and Execution: Input & Output for All Possible Cases	3		
4	**Viva Voce**	Program Analysis & Applications	2		
5		Communication & Confidence Level	1		
		TOTAL	10		
		Signature	☒		

Program No. 2

Paste your DATA SHEET here

[Page intentionally left blank, student to paste his/her data sheet after evaluated by the staff in-charge]

Program No. 3

Write a Java application to insert data into any DATABASE and retrieve info based on particular queries (For example update, delete, search etc...) with proper usage of user defined Exception Handling (also make use of throw, throws.) (Ex: if input exceeds the greater than the text field then user defined exception can be given)

PART 3.A

Part A: Write a Java application to insert data into any DATABASE and retrieve info based on particular queries (For example update, delete, search etc...)

Question: Write a JAVA Program to insert data into Student database and retrieve info based on particular queries (For example update, delete, search etc…).

Step wise Execution Procedure:

Step 1: Create a class with student in that methods included are:

Step 1.1: Create a method connection which returns the Connection object. The proper use of SQLException for handling different kind of errors with proper printing of messages.

Step 1.2: Create a method insert which accepts the arguments like usn, name, address, phoneno, sem, section, department.

Step 1.2.1: calling the connection object and create a statement.

Step 1.3.1: Execute of the insert query with arguments to be done.

Step 1.4.1: Closing of Connection with proper handling of the SQLException and message

Step 1.3: Create a method display.

Step 1.3.1: proper display of student details to be done.

Step 1.3.2: Proper handling of the SQLException and message

Step 1.4: Create a method edit which accepts the argument usn and name.

Step 1.4.1: Calling of Connection method and create a statement.

Step 1.4.2: Execute of the edit query with the arguments like name will be changed based on the usn.

Step 1.4.3: closing of the Connection with proper handling of the SQLException and with message.

Step 1.5: Create a method delete which accepts the argument usn.

Step 1.4.1: Calling of Connection method and create a statement.

Step 1.4.2: Execute of the delete query with the argument of usn and deletion of respective usn should be done.

Step 1.4.3: closing of the Connection with proper handling of the SQLException and with message.

Step 1.6: Create a method main.

Step 1.5.1: A menu driven program to be designed for proper input from the user and execution of all the above said CRUD methods.

Expected Output:

A menu driven program to be displayed with all the CRUD methods.

PART 3.B

Part B: Write a Java application to insert data into any DATABASE and retrieve info based on particular queries (For example update, delete, search etc...) with proper usage of user defined Exception Handling (also make use of throw, throws.) (Ex: if input exceeds the greater than the text field then user defined exception can be given)

Question: Write a Java application to insert data into any DATABASE and retrieve info based on particular queries (For example update, delete, search etc...), if input exceeds the greater than the text field then user defined exception can be given)

Step wise Execution Procedure:

Step 1: Create a class with student in that methods included are:

Step 1.1: Create a method connection which returns the Connection object. The proper use of SQLException for handling different kind of errors with proper printing of messages.

Step 1.2: Create a method insert which accepts the arguments like usn, name, address, phoneno, sem, section, department.

Step 1.2.1: calling the connection object and create a statement.

Step 1.3.1: Execute of the insert query with arguments to be done.

Step 1.4.1: Closing of Connection with proper handling of the SQLException and message

Step 1.3: Create a method display.

Step 1.3.1: proper display of student details to be done.

Step 1.3.2: Proper handling of the SQLException and message

Step 1.4: Create a method edit which accepts the argument usn and name.

Step 1.4.1: Calling of Connection method and create a statement.

Step 1.4.2: Execute of the edit query with the arguments like name will be changed based on the USN.

Step 1.4.3: closing of the Connection with proper handling of the SQLException and with message.

Step 1.5: Create a method delete which accepts the argument usn.

Step 1.4.1: Calling of Connection method and create a statement.

Step 1.4.2: Execute of the delete query with the argument of usn and deletion of respective usn should be done.

Step 1.4.3: closing of the Connection with proper handling of the SQLException and with message.

Step 1.6: Create a method which displays the customized exception during the input of number

Step 1.7: Create a method main.

Step 1.5.1: A menu driven program to be designed for input from the user and execution of all the above said CRUD methods

Step 1.5.2: During the input if the input exceed the size then and exception to be thrown to display the customized message

Expected Output:

Output should display the crud operations on the database, if exception occur during the input customized message has to be shown.

Sl.No	Rubrics for Practice Sessions		Max. Marks	Marks Obtained 3.a	Marks Obtained 3.b
1	**Conduction & Execution**	Understanding of Concepts and Logic	2		
2		Approach Towards the Problem	2		
3		Conduction and Execution: Input & Output for All Possible Cases	3		
4	**Viva Voce**	Program Analysis & Applications	2		
5		Communication & Confidence Level	1		
		TOTAL	10		
		Signature	☒		

Program No. 3

Paste your DATA SHEET here

[Page intentionally left blank, student to paste his/her data sheet after evaluated by the staff in-charge]

Program No. 4

Write a Java Servlet Program to implement a dynamic HTML for the following scenario

a. Display the auto web page refresh using HTTPRequest and HTTPResponse

b. Verify the user authentication using ServletConfig

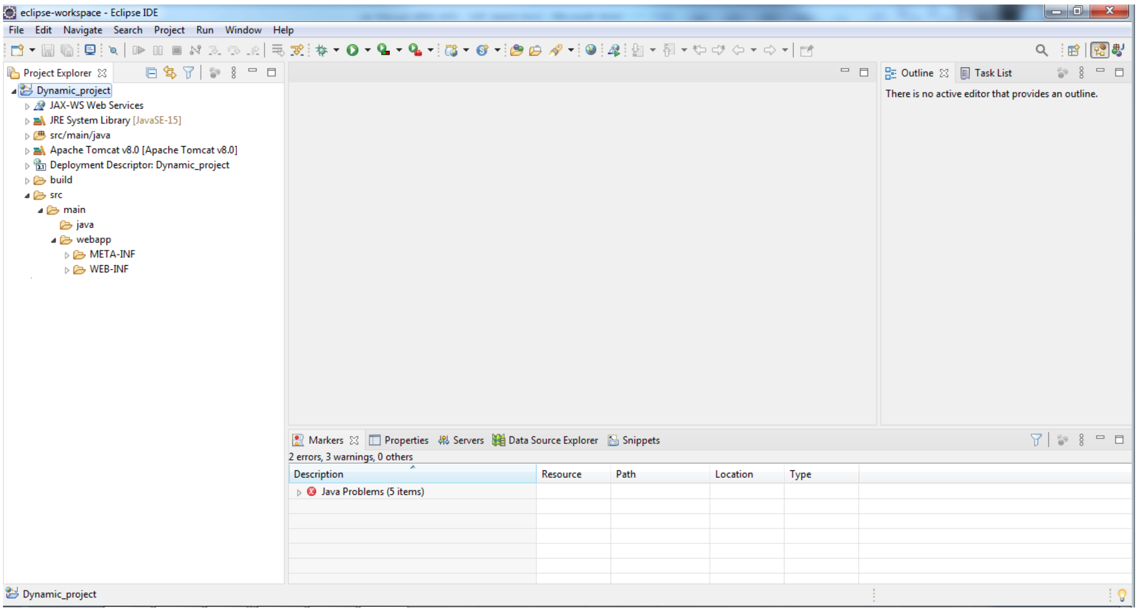
PART 4.A

Part A Write a Java Servlet Program to implement a dynamic HTML for display the auto web page refresh using HTTPRequest and HTTPResponse

Question: Write a JAVA Servlet Program: Consider a webpage which is displaying Date and time or stock market status. For all such type of pages, you would need to refresh your web page regularly; Java Servlet makes this job easy by providing refresh automatically after a given interval

Initial Steps for creating the project in Eclipse IDE: (these steps are same for programs 5, 6)

- **Step 1:** Create a Java Project in Eclipse IDE
  - Select - File → New → Project
  - Select from the list – Web --> Dynamic Web Project
- **Step 2:** Click on next button
- **Step 3:** select the proper **target runtime – apache tomcat version**
- **Step 4:** Provide the suitable Project Name, click on **finish** button
  (Note: do not provide the java keywords as project name)
- **Step 5:** Project will be created as shown in the Eclipse IDE
  The figure shows the project tab at the left



```
Dynamic_project
  JAX-WS Web Services
  JRE System Library [JavaSE-15]
  src/main/java
  Apache Tomcat v8.0 [Apache Tomcat v8.0]
  Deployment Descriptor: Dynamic_project
  build
  src
    main
      java
      webapp
        META-INF
        WEB-INF
```

There is no active editor that provides an outline.

2 errors, 3 warnings, 0 others

Description	Resource	Path	Location	Type
Java Problems (5 items)				

Dynamic_project

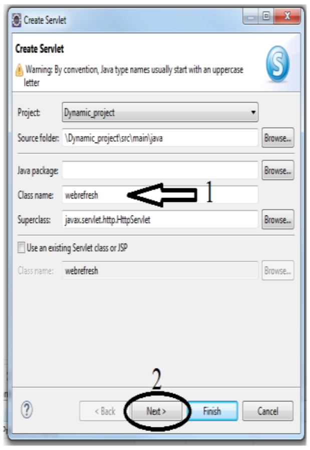
Fig: Shows the Eclipse IDE for new dynamic web project creation

Step wise Execution Procedure:

Step 1: create a Servlet inside the main project src

Right click on the web application → click on new → Servlet (A wizard will pop up)

Step 2: Provide the Servlet Name and Package name properly. Click Next button (Note: do not provide the java keywords as Servlet and package name). Click on check box on add information to deployment descriptor (web.xml) → click on finish



Create Servlet

Warning: By convention, Java type names usually start with an uppercase letter

Project: Dynamic_project

Source folder: \Dynamic_project\src\main\java

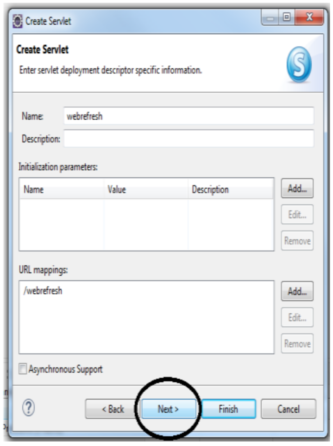
Java package:

Class name: webrefresh

Superclass: javax.servlet.http.HttpServlet

□ Use an existing Servlet class or JSP

Class name: webrefresh



Create Servlet

Enter servlet deployment descriptor specific information.

Name: webrefresh

Description:

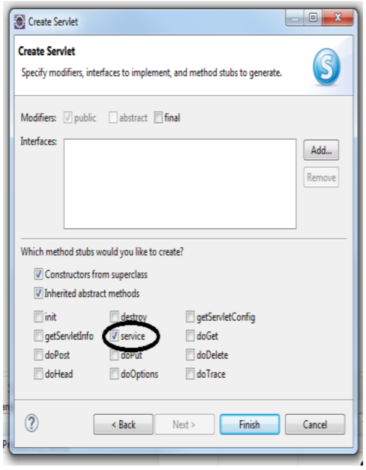
Initialization parameters:

Name	Value	Description

URL mappings:

/webrefresh

□ Asynchronous Support



Create Servlet

Specify modifiers, interfaces to implement, and method stubs to generate.

Modifiers: ☑ public □ abstract □ final

Interfaces:

Which method stubs would you like to create?

☑ Constructors from superclass

☑ Inherited abstract methods

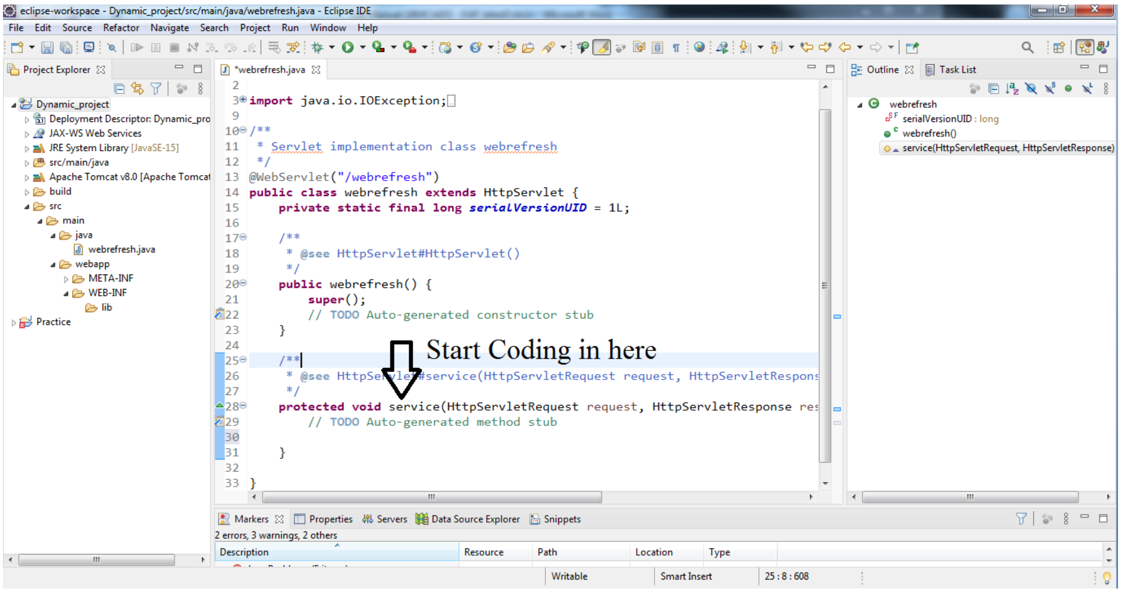
□ init □ destroy □ getServletConfig

□ getServletInfo ☑ service □ doGet

□ doPost □ doPut □ doDelete

□ doHead □ doOptions □ doTrace

Fig 1: Add the ClassName Fig 2: Click on Next Fig 3: select Service method



eclipse-workspace - Dynamic_project/src/main/java/webrefresh.java - Eclipse IDE

*webrefresh.java

```
import java.io.IOException;

/**
 * Servlet implementation class webrefresh
 */
@WebServlet("/webrefresh")
public class webrefresh extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public webrefresh() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletRespons
     */
    protected void service(HttpServletRequest request, HttpServletResponse res
        // TODO Auto-generated method stub

    }

}
```

2 errors, 3 warnings, 2 others

Fig 4: The file will be created

Step 3: In the Servlet in service method page has to refresh for every stipulated time using HttpServletResponse interface.
For example:
// Set refresh, autoload time as 5 seconds

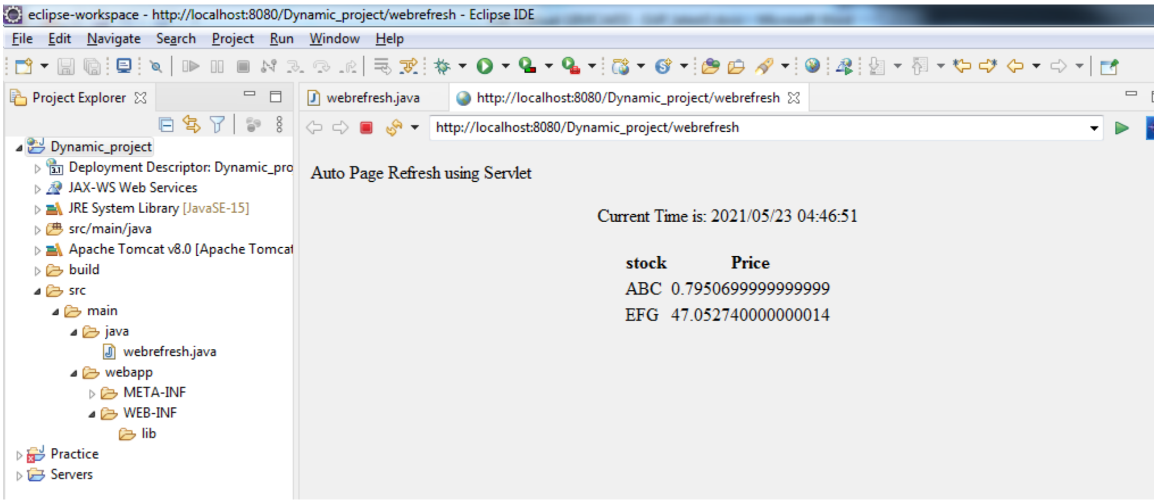
```
response.setIntHeader(“Refresh”, 5);

// Get current time
DateFormat dateFormat = new SimpleDateFormat(“yyyy/MM/dd HH:mm:ss”);
Date date = new Date();
price1 = price1 * .43;
price2 = price2 * 1.33;
PrintWriter out = response.getWriter();
String title = “Auto Page Refresh using Servlet”;
out.println( “<html>\n” + “<body bgcolor=\”#f0f0f0\”>\n” +
“<h1align=\”center\”>” + title + “</h1>\n” +
“<p align =\”center\”>Current Time is: “ + dateFormat.format(date) + p>”+
“<p align =\”center\”><table><tr><th>stock</th><th>Price</th></tr>”+
“<tr><td>ABC</td><td>”+price1+”</td></tr>
<tr><td> EFG</td><td>”+price2+”</td></tr></table></p>”
);
```

Step 4: Deploy and run project on server

Step 5: For execution of the code, open the browser and type the url

http://localhost:8080/project_name/Servlet_name

Expected Output:


eclipse-workspace - http://localhost:8080/Dynamic_project/webrefresh - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
webrefresh.java
http://localhost:8080/Dynamic_project/webrefresh
Dynamic_project
Deployment Descriptor: Dynamic_pro
JAX-WS Web Services
JRE System Library [JavaSE-15]
src/main/java
Apache Tomcat v8.0 [Apache Tomcat
build
src
main
java
webrefresh.java
webapp
META-INF
WEB-INF
lib
Practice
Servers

```
Auto Page Refresh using Servlet

Current Time is: 2021/05/23 04:46:51

stock      Price
ABC  0.7950699999999999
EFG  47.052740000000014
```

Fig : Page showing the Auto page refresh servlet

PART 4.B

Part B Write a Java Servlet Program to implement a dynamic HTML for Verify the user authentication using ServletConfig

Question: Write a Java Servlet Program to implement a dynamic HTML for following scenarios

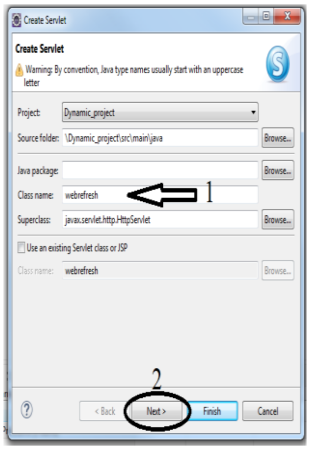
a. User name and password should be accepted in HTML
b. Verify the username and password using a ServletConfig initparam and display the appropriate message on another Servlet

Step wise Execution Procedure:

Step 1: create a Servlet inside the main project src

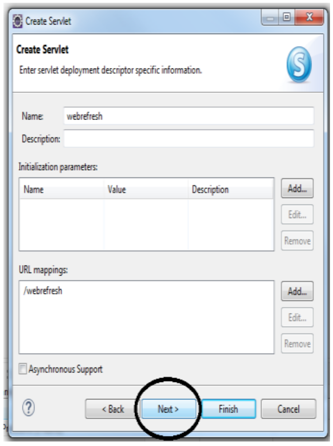
Right click on the web application → click on new → Servlet (A wizard will pop up)

Step 2: Provide the Servlet Name and Package name properly. Click Next button (Note: do not provide the java keywords as Servlet and package name). Click on check box on add information to deployment descriptor (web.xml) → click on finish


Create Servlet

Warning: By convention, Java type names usually start with an uppercase letter

Project: Dynamic_project
Source folder: \Dynamic_project\src\main\java
Java package:
Class name: webrefresh
Superclass: javax.servlet.http.HttpServlet
□ Use an existing Servlet class or JSP
Class name: webrefresh

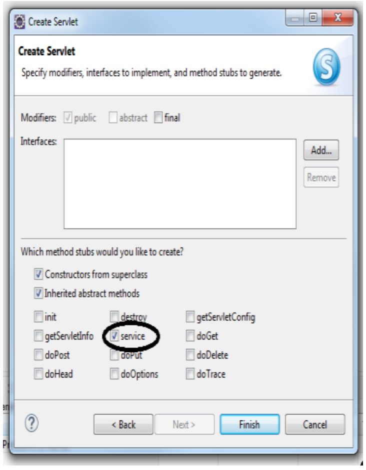

Create Servlet

Enter servlet deployment descriptor specific information.

Name: webrefresh
Description:
Initialization parameters:

Name	Value	Description

URL mappings: /webrefresh
□ Asynchronous Support

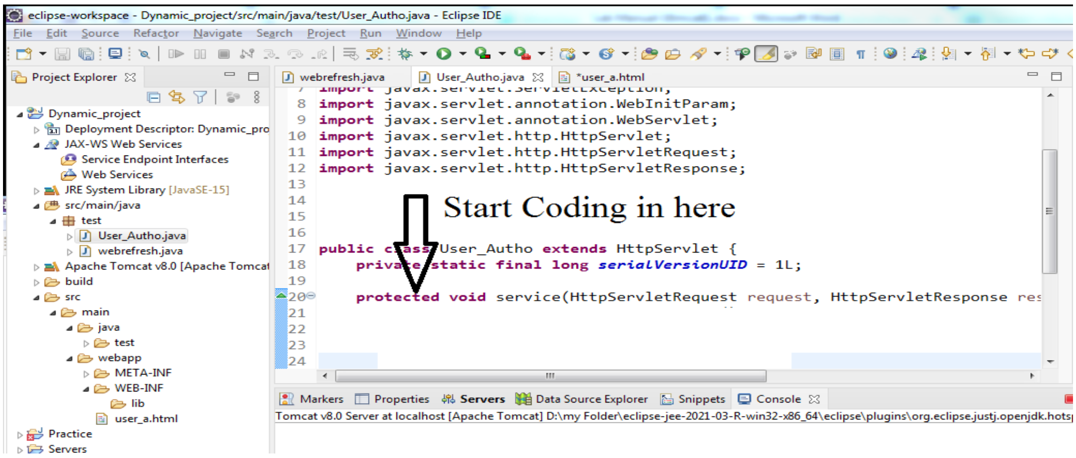

Create Servlet

Specify modifiers, interfaces to implement, and method stubs to generate.

Modifiers: ☑ public □ abstract □ final
Interfaces:

Which method stubs would you like to create?
☑ Constructors from superclass
☑ Inherited abstract methods
□ init □ destroy □ getServletConfig
□ getServletInfo ☑ service □ doGet
□ doPost □ doPut □ doDelete
□ doHead □ doOptions □ doTrace

Fig 1: Add the ClassName Fig 2: Click on Next Fig 3: select Service method

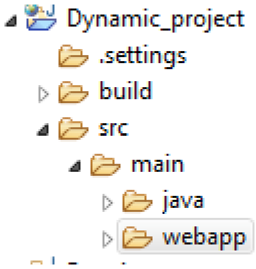

eclipse-workspace - Dynamic_project/src/main/java/test/User_Autho.java - Eclipse IDE

```
import javax.servlet.ServletException;
import javax.servlet.annotation.WebInitParam;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


public class User_Autho extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void service(HttpServletRequest request, HttpServletResponse res
```

Tomcat v8.0 Server at localhost [Apache Tomcat] D:\my Folder\eclipse-jee-2021-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hots

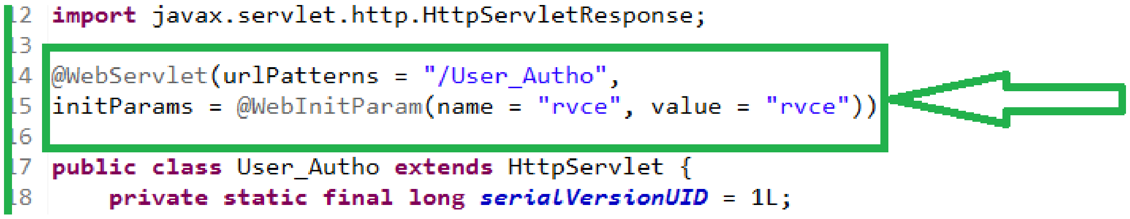
Fig 4: The file will be created

Step 3: Create an HTML file under the **webapp** folder and add the code for accepting username and password with submit button. In the <form> tag add the WebServlet url pattern.



```
Dynamic_project
    .settings
    build
    src
        main
            java
            webapp
```

Step 4: Add the Annotation of **WebInitParam** inside the **WebServlet** as shown below



```
import javax.servlet.http.HttpServletResponse;

@WebServlet(urlPatterns = "/User_Autho",
initParams = @WebInitParam(name = "rvce", value = "rvce"))

public class User_Autho extends HttpServlet {
    private static final long serialVersionUID = 1L;
```

Fig: Shows the WebServlet annotation

Step 5: Accept the Username and Password from the html and compare with the servletconfig and display the result

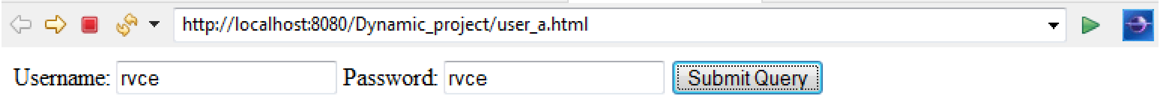
Example code:

```
PrintWriter pw = response.getWriter();
ServletConfig sc = getServletConfig();
String pass = sc.getInitParameter("rvce");
String uname = request.getParameter("username");
String passw = request.getParameter("password");
if(uname.equalsIgnoreCase("rvce") && passw.equalsIgnoreCase(pass)) {
        pw.println(" username and password is valid ");
}else {
        pw.println(" username and password is invalid ");
}
```

Step 5: For execution of the code, open the browser and type the url

http://localhost:8080/project_name/Servlet_name

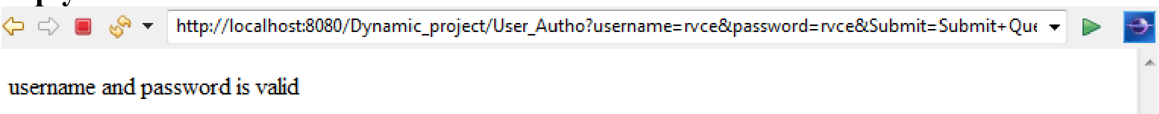
Expected Output:
Input:



http://localhost:8080/Dynamic_project/user_a.html

Username: rvce Password: rvce

Reply from the server:



http://localhost:8080/Dynamic_project/User_Autho?username=rvce&password=rvce&Submit=Submit+Que

username and password is valid

Program No. 4

Paste your DATA SHEET here

[Page intentionally left blank, student to paste his/her data sheet after evaluated by the staff in-charge]

Sl.No	Rubrics for Practice Sessions		Max. Marks	Marks Obtained 4.a	Marks Obtained 4.b
1	**Conduction & Execution**	Understanding of Concepts and Logic	2		
2		Approach Towards the Problem	2		
3		Conduction and Execution: Input & Output for All Possible Cases	3		
4	**Viva Voce**	Program Analysis & Applications	2		
5		Communication & Confidence Level	1		
		TOTAL	10		
		Signature			

Program No. 5

Write a JSP program to demonstrate for a given scenario

a. Session API and cookies

b. Conditional and exceptional handling techniques

PART 5.A

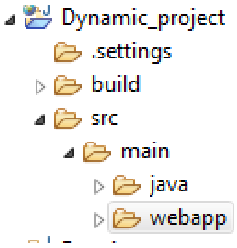
Part A Write a JSP program to demonstrate Session API and Cookies

Question: Write a JSP program to

a. Display the methods in Session API
b. Remember the user preferences using cookies

Setup wise Execution Procedure:

Step 1: Create a Dynamic Web Project, Create an HTML and JSP file under the **webapp** folder and add the code for accepting username and password with submit button.



Step 2: Add the code in html which accept the user information like First name and Last Name with an submit button. In the <form> tag add JSP page name in action attribute.

Step 3: Add the code in JSP page to display the session API methods and accept the parameters from the html file and store in cookies. Display the stored cookies in the same page.

Example: **Storing cookies**

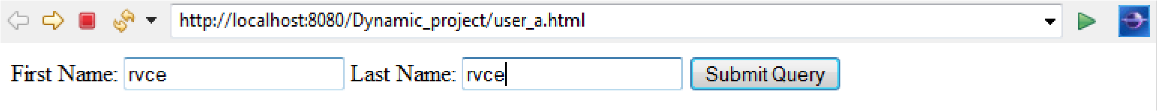
- Cookie c = new Cookie("userID", "a1234");
- Set the maximum age.
- To tell browser to store cookie on disk instead of just in memory, use setMaxAge (argument is in seconds)
- c.setMaxAge(60*60*24*7); // One week
- Place the Cookie into the HTTP response, Use **response.addCookie**.
- If you forget this step, no cookie is sent to the browser! **response.addCookie(c);**
- Reading the Cookies back:

```
Cookie[] cookies = request.getCookies();
   if (cookies != null) {
       for(Cookie cookie: cookies) {
          if (cookieName.equals(cookie.getName())) {
             doSomethingWith(cookie.getValue());
          }
       }
   }
```

Step 4: Deploy the project and run on server, .html as the first program to execute

Expected output:

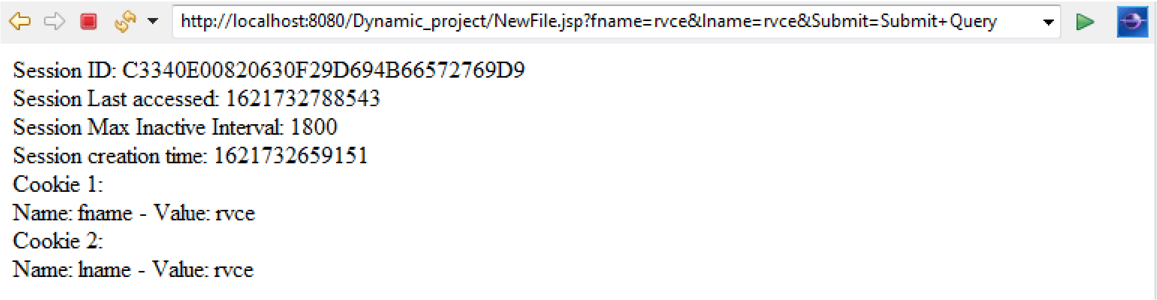
Input Screen:



http://localhost:8080/Dynamic_project/user_a.html

First Name: rvce Last Name: rvce

Reply from Server:



http://localhost:8080/Dynamic_project/NewFile.jsp?fname=rvce&lname=rvce&Submit=Submit+Query

```
Session ID: C3340E00820630F29D694B66572769D9
Session Last accessed: 1621732788543
Session Max Inactive Interval: 1800
Session creation time: 1621732659151
Cookie 1:
Name: fname - Value: rvce
Cookie 2:
Name: lname - Value: rvce
```

PART 5.B

Part B Write a JSP program to demonstrate Conditional and Exception Handling

Question: Write a JSP program to demonstrate user authentication process. If the user input is blank throw an exception.

Step1: Create a Dynamic Web Project in Eclipse IDE, create an HTML file under the **webapp** folder with First name, Last name and age. Add the JSP file name in action attribute of <form> tag with a submit button inside the <form> tag.

Step 2: Create a JSP page which accepts the parameter values from the HTML page and check if any of the parameter is empty or not. If the parameters accepted is empty then throw an exception and display the exception in another page, if not display all the parameters in the same page.

Example:

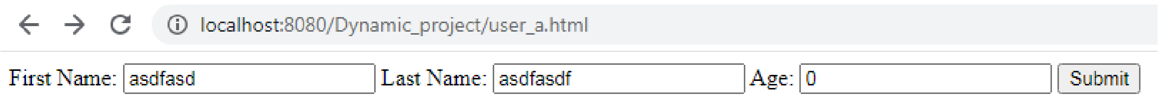
```
<html>
<body>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" errorPage="error1.jsp"%>

<%-- Accept the parameters --%>
<%String fname = request.getParameter("fname");
String lname = request.getParameter("lname");
int age = Integer.parseInt(request.getParameter("age"));%>

<%-- conditional statements --%>
<%if(fname.isBlank() || lname.isBlank() || age==0){

        throw new Exception("Fields are empty");

}else{%>

First Name: <%=fname %><br>
Last Name: <%=lname %><br>
Age: <%=age %><br>
<%} %>

</body>
</html>
```

Step 3: Deploy and run on server taking the html page as first file to execute.

Expected Output:

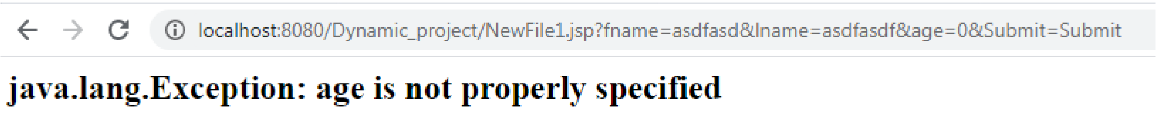
Input:



localhost:8080/Dynamic_project/user_a.html

First Name: asdfasd Last Name: asdfasdf Age: 0

Reply from the server:



localhost:8080/Dynamic_project/NewFile1.jsp?fname=asdfasd&lname=asdfasdf&age=0&Submit=Submit

java.lang.Exception: age is not properly specified

Sl.No	Rubrics for Practice Sessions		Max. Marks	Marks Obtained 5.a	Marks Obtained 5.b
1	**Conduction & Execution**	Understanding of Concepts and Logic	2		
2		Approach Towards the Problem	2		
3		Conduction and Execution: Input & Output for All Possible Cases	3		
4	**Viva Voce**	Program Analysis & Applications	2		
5		Communication & Confidence Level	1		
		TOTAL	10		
		Signature			

Program No. 5

Paste your DATA SHEET here

[Page intentionally left blank, student to paste his/her data sheet after evaluated by the staff in-charge]

APPENDIX A

QUESTION BANK

1. What is the difference between a constructor and a method?
2. What is the purpose of garbage collection in Java, and when is it used?
3. Describe synchronization in respect to multithreading.
4. What is an abstract class?
5. What is the difference between an Interface and an Abstract class?
6. Explain different way of using thread?
7. State the significance of public, private, protected, default modifiers both singly and in combination and state the effect of package relationships on declared items qualified by these modifiers.
8. What is static in java?
9. What is final class?
10. What if the main() method is declared as private?
11. What if the static modifier is removed from the signature of the main() method?
12. What are Checked and UnChecked Exception?
13. What is Overriding?
14. What is the default value of an object reference declared as an instance variable?
15. Can a top level class be private or protected?
16. What type of parameter passing does Java support?
17. Primitive data types are passed by reference or pass by value?
18. Objects are passed by value or by reference?
19. What are wrapper classes?
20. Why do we need wrapper classes?
21. What are checked exceptions
22. What are runtime exceptions?
23. What is the difference between error and an exception?
24. How to create custom exceptions?
25. Is it necessary that each try block must be followed by a catch block?
26. If I write return at the end of the try block, will the finally block still execute?
27. What is synchronization and why is it important?
28. What is the purpose of finalization?
29. What is the difference between static and non-static variables?
30. How are this() and super() used with constructors?
31. How is final different from finally and finalize()?
32. What is the most important feature of Java?
33. What do you mean by platform independence?
34. What is a JVM?
35. Are JVM's platform independent?
36. What is the difference between a JDK and a JVM?
37. What is a pointer and does Java support pointers?
38. What is the base class of all classes?
39. Does Java support multiple inheritance?
40. Is Java a pure object oriented language?
41. Are arrays primitive data types?
42. What is difference between Path and Classpath?
43. What are local variables?
44. What is a package?
45. Which package is imported by default?

46. What is difference between Array and ArrayList? When will you use Array over ArrayList?
47. What is a Servlet?
48. What's the advantages using Servlets over using CGI?
49. What are the general advantages and selling points of Servlets?
50. Which package provides interfaces and classes for writing Servlets?
51. What's the Servlet Interface?
52. When a Servlet accepts a call from a client, it receives two objects. What are they?
53. What information does ServletRequest allow access to?
54. What type of constraints can ServletResponse interface set on the client?
55. Explain Servlet lifecycle?
56. How does HTTP Servlet handle client requests?
57. Differentiate between the ServletContext and ServletConfig objects.
58. How is the Get method different from the Post method?
59. What is a session?
60. What is a cookie?
61. What is the difference between GenericServlet and HttpServlet?
62. What is JSP?
63. What are advantages of using JSP?
64. What are the advantages of JSP over Pure Servlets?
65. What are the advantages of JSP over Server-Side Includes (SSI)?
66. What are the stages of the JSP Lifecycle?
67. What is a sciptlet in JSP and what is its syntax?
68. What are JSP declarations?
69. What are JSP expressions?
70. What are JSP comments?
71. What are JSP Directives?
72. What are the types of directive tags?
73. What are JSP actions?
74. Name some JSP actions.
75. What is a page directive?
76. What are various attributes Of page directive?
77. What is extends attribute?
78. What is a taglib directive?
79. What is the function of <jsp:include> action?