

**MAWLANA BHASHANI SCIENCE AND TECHNOLOGY
UNIVERSITY
SANTOSH, TANGAIL-1902**



**DEPARTMENT OF INFORMATION AND COMMUNICATION
TECHNOLOGY**
Lab Report

Lab Report No : 01

Lab Report on : Compare abstract classes and interfaces in terms of multiple inheritance.

Course Title : Software Engineering and Project Management

Course Code : ICT-3108

Submitted By	Submitted To
Name: Md. Nadim Mahmud Borad ID: IT23012 3rd Year, 1st Semester Session: 2022-2023 Dept. of ICT, MBSTU	Dr. Ziaur Rahman Professor Department of Information and Communication Technology Mawlana Bhashani Science and Technology University

Lab report no -1 [2]

Lab report name- Comparison of abstract Class and interfaces in terms of multiple inheritance.

Objective - The objective of the lab is to compare abstract classes and interfaces in java with respect of multiple inheritance.

Introduction - Java is a object oriented language that support inheritance to promote code reusability .

Abstract class in java - An abstract class is a class that can't be instantiated and may contain abstract methodologies.

code - abstract class Animal {
 abstract void sound();
 void sleep();
 System.out.println("Animal is
 sleeping");
}

```
Class Dog extend Animal{  
    void sound(){  
        System.out.println("Dog barks");  
    }  
}
```

Discussion - Abstract classes provide partial abstraction and are useful when sharing code among related classes. They restrict inheritance to one class only. Interfaces provide full abstraction and allow a class to inherit from multiple interfaces, making them more flexible and suitable for large scale applications.

**MAWLANA BHASHANI SCIENCE AND TECHNOLOGY
UNIVERSITY
SANTOSH, TANGAIL-1902**



**DEPARTMENT OF INFORMATION AND COMMUNICATION
TECHNOLOGY**
Lab Report

Lab Report No : 02

Lab Report on : How does encapsulation ensure data security and integrity.

Course Title : Software Engineering and Project Management

Course Code : ICT-3108

Submitted By	Submitted To
Name: Md. Nadim Mahmud Borad ID: IT23012 3rd Year, 1st Semester Session: 2022-2023 Dept. of ICT, MBSTU	Dr. Ziaur Rahman Professor Department of Information and Communication Technology Mawlana Bhashani Science and Technology University

Lab report : 2 [3]

Encapsulation is an oop concept that hide data and protected them from direct access. Which ensure data security and data integrity. Encapsulation hide data using private variable. This method allow validation logic inside setter methods. Ensure the object is always in a valid state and also can reject invalid input.

Code - BankAccount.java

```
public class BankAccount {  
    private String accountNo;  
    private double balance;  
    public void setAccountNo (String AccNo)  
    { if (AccountNo == null) {  
        System.out.println ("Invalid!");  
        return;  
    } }
```

```
this.accountNo = accountNo;
}
public void setIntBalance(double balance) {
    if (balance < 0) {
        System.out.println("Invalid!");
    }
    this.balance = balance;
}
public String getAccountNo() {
    return accountNo;
}
public double getBalance() {
    return balance;
}
```

**MAWLANA BHASHANI SCIENCE AND TECHNOLOGY
UNIVERSITY
SANTOSH, TANGAIL-1902**



**DEPARTMENT OF INFORMATION AND COMMUNICATION
TECHNOLOGY
Lab Report**

Lab Report No : 04

Lab Report on : how JDBC manages communication between a Java application and a relational database.

Course Title : Software Engineering and Project Management

Course Code : ICT-3108

Submitted By	Submitted To
Name: Md. Nadim Mahmud Borad ID: IT23012 3rd Year, 1st Semester Session: 2022-2023 Dept. of ICT, MBSTU	Dr. Ziaur Rahman Professor Department of Information and Communication Technology Mawlana Bhashani Science and Technology University

Lab report 4

JDBC stand for Java Database connection which is used to connect a java program with a relational database.

JDBC works as a bridge between Java application and database.

The Java program sends SQL queries using JDBC. JDBC driver receives the request and gives to database.

After a SQL query is written, the query is executed using executeQuery method.

Java Code -

```
Import java.sql.*;
```

```
class selectExample{
```

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

```
collection con = null;
```

```
try {
```

```
    con = prentManager.getconnection()  
    "Jdbc:mysql://localhost:3306/TestDB;root,"  
    "password");
```

```
statement st = con.createStatement();
```

```
Resultset rs = st.executeQuery("SELECT * From student");
```

```
while (rs.next()) {
```

```
    System.out.println(rs.getInt(1) + "+" +
```

```
    rs.getString(2));
```

```
}
```

```
}
```

```
catch (Exception e) {
```

```
    System.out.println("Error");
```

```
} finally { try { if (con != null) con.close(); } }
```

```
catch (Exception e) {} }
```

**MAWLANA BHASHANI SCIENCE AND TECHNOLOGY
UNIVERSITY
SANTOSH, TANGAIL-1902**



**DEPARTMENT OF INFORMATION AND COMMUNICATION
TECHNOLOGY**
Lab Report

Lab Report No : 05

Lab Report on : Java EE application and a servlet controller manage

Course Title : Software Engineering and Project Management

Course Code : ICT-3108

Submitted By	Submitted To
Name: Md. Nadim Mahmud Borad ID: IT23012 3rd Year, 1st Semester Session: 2022-2023 Dept. of ICT, MBSTU	Dr. Ziaur Rahman Professor Department of Information and Communication Technology Mawlana Bhashani Science and Technology University

Lab report 5:

In a Java EE application, a servlet works as a controller. The controller manages the flow between model and view. The model contains business data and logic. The servlet get data from the model. Then it send this data to view JSP is used as the view to show data to the user.

Java code -

```
import java.io.*  
import javax.servlet.*  
import javax.servlet.http.*
```

```
public class helloServlet extends HttpServlet {  
    protected void doGet(HttpServletRequest request,  
                         HttpServletResponse response) throws  
        ServletException, IOException {  
        String name = "student";  
        request.setAttribute("msg", name);  
        RequestDispatcher rd = request.getRequestDispatcher(  
            "Hello.java");  
        rd.forward(request, response);  
    }  
}
```

JSP code view (HTML) :

```
<html>  
  <body>  
    <h2> Hello, ${msg} </h2>  
  </body>  
</html>
```

**MAWLANA BHASHANI SCIENCE AND TECHNOLOGY
UNIVERSITY
SANTOSH, TANGAIL-1902**



**DEPARTMENT OF INFORMATION AND COMMUNICATION
TECHNOLOGY**
Lab Report

Lab Report No : 06

Lab Report on : How does PreparedStatement improve performance and security over Statement in JDBC.

Course Title : Software Engineering and Project Management

Course Code : ICT-3108

Submitted By	Submitted To
Name: Md. Nadim Mahmud Borad ID: IT23012 3rd Year, 1st Semester Session: 2022-2023 Dept. of ICT, MBSTU	Dr. Ziaur Rahman Professor Department of Information and Communication Technology Mawlana Bhashani Science and Technology University

Lab report : 6

Prepared Statement improves security by preventing SQL injection because SQL queries are precompiled with placeholders (?) and user input is treated strictly as data, not executable SQL. It also improves performance because the SQL statement is compiled once by the database and can be reused multiple times with different values, reducing parsing and execution overhead compared to statements with recompiles the query every time.

Java code:

```
import class InsertExample;
public static void main(String[] args) throws Exception {
    String url = "Jdbc:mysql://localhost:3306/testbd";
    String user = "root";
    String pass = "123";
    Connect con = DriverManager.getConnection(url, user, pass);
    String sql = "Insert into student(id, name, age) value(?, ?, ?)";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setInt(1, 1);
    ps.setString(2, "Nordin");
    ps.executeUpdate();
}
```

1) Parent.java ,

```
package mypk;
public class Parent {
    protected int num = 10;
}
```

Child.java

```
package mypk;
public class Child extends Parent {
    public void show() {
        System.out.println("Number = " + num);
    }
    public static void main(String[] args) {
        Child obj = new Child();
        obj.show();
    }
}
```

num is protected but as the child
is in the same package so it
can directly access it .