|  |  |
| --- | --- |
| *A close up of a logo  Description automatically generated* | *DEPARTMENT OF COMPUTER ENGINEERING* |

|  |  |
| --- | --- |
| Semester | S.E. Semester III – Computer Engineering |
| Subject | Object Oriented Programming Using Java (Skill Based Lab) |
| Subject Professor In-charge | Prof. Indu Anoop |
| Laboratory | Online Lab |

|  |  |  |
| --- | --- | --- |
| Student Name | Trisha Shah | |
| Roll Number | 20102A0004 | |
| Grade and Subject Teacher’s Signature |  |  |

|  |  |  |
| --- | --- | --- |
| Experiment | 14 | |
| Problem Statement | WAP on multithreading | |
| Resources / Apparatus Required | Hardware: Computer System | Software: jdk 1.8, Eclipse / Notepad++/IntelliJ IDEA |
| Details | **What is thread :-**  A thread is a light weight process.  A thread is a subpart of a process that can run individually.  In java, a thread goes through different states throughout its execution. These stages are called thread life cycle  The thread base is an instantiation of a class which either implements Runnable extends Thread  **Thread class:**  Thread class provide constructors and methods to create and perform operations on a thread. Thread class extends Object class and implements Runnable interface.  **Runnable interface:**  The Runnable interface should be implemented by any class whose instances are intended to be executed by a thread. Runnable interface have only one method named run() | |
| Code | class Account{  private double balance;  private int accno;  public Account(double balance,int accno){  this.balance=balance;  this.accno=accno;  }  void chkb(){  System.out.println("Account no: " + accno+ "Balance is " +balance);  }  void deposit(double amt)  {  this.balance+=amt;  }  void withdraw(double amt)  {  this.balance-=amt;  }  }  class Accountholder implements Runnable{  Account account;  public Accountholder(Account account){  this.account=account;  }  public void run(){  this.account.deposit(100.0);  this.account.withdraw(25.0);  this.account.deposit(75.0);  this.account.withdraw(100.0);  this.account.chkb();  }  }  public class P2{  public static void main(String[] args){  Account acc= new Account(25.0,111);  Accountholder joe= new Accountholder(acc);  Accountholder john= new Accountholder(acc);  Thread t1= new Thread(joe);  Thread t2= new Thread (john);  t1.start();  t2.start();  }  } | |
| Output |  | |
| Conclusion | We successfully executed program to implement Multi-threading | |