|  |  |
| --- | --- |
| *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 | 13 | |
| Problem Statement | WAP to demonstrate User Defined Exception | |
| Resources / Apparatus Required | Hardware: Computer System | Software: jdk 1.8, Eclipse / Notepad++/IntelliJ IDEA |
| Details | User Defined Exception or custom exception is creating your own exception class and throw that exception using ‘throw’ keyword. This can be done by extending the class Exception which makes it a checked Exception Type or by extending the RuntimeExceptionClass which makes it an unchecked Exception | |
| Code | import java.util.Scanner;  import java.util.regex.Pattern;  class InvalidPasswordException extends Exception{  int passwordConditionViolated;  public InvalidPasswordException(int conditionViolated) {  passwordConditionViolated=conditionViolated;  }  public String printCustomMessage() {  switch(passwordConditionViolated) {  case 1: return("Password Length should be 8 characters");  case 2: return("Password should contain at least one digit");  case 3: return("Password should contain at least one special symbol");  }  return ("");  }  }  class PasswordValidatorExceptionDemo{  public static void main(String[] args){  System.out.println("Enter your username");  Scanner in=new Scanner(System.in);  String username=in.nextLine();  System.out.println("Enter your password");  String password=in.nextLine();  try {  System.out.println("Is the"+password+" password valid?");  isValid(password);  System.out.println("Login Successful");  } catch (InvalidPasswordException e) {  System.out.println(e.printCustomMessage());  }    }  private static void isValid(String password) throws InvalidPasswordException {    if(password.length()<8) {  throw new InvalidPasswordException(1);    }  if(true) {  int count=0;  //check digits from 0 to 9  for (int i=0;i<=9;i++) {  String str1=Integer.toString(i);  if(password.contains(str1)) {  count=1;  }  }  if(count==0) {  throw new InvalidPasswordException(2);  }  }  Pattern specialCharPattern=Pattern.compile("[^a-z0-9]", Pattern.CASE\_INSENSITIVE);  if(!specialCharPattern.matcher(password).find()) {  throw new InvalidPasswordException(3);  }    }  } | |
| Output |  | |
| Conclusion | Thus students could explore exceptions in the program | |