**EXPT: 1**

**DATE:**

**APPLICATION USING CLASSES AND OBJECTS**

**AIM:**

To develop a java application to generate the electricity bill.

**ALGORITHM:**

Step 1: Start the netbeansIDE8.2

Step 2: Goto FILE🡪 new project🡪java🡪java application🡪click next

Step 3: Give the name for the application🡪Click Finish.

Step 4: Create a class names EBbillCalculation with the necessary variables.

Step 5: Create getdetails() which gets the basic information form the user

Step 6: Create two more method Domestic\_Calculation &

Commerical\_Calculation to calculate the bill and print it.

Step 7: Stop the program.

**PROGRAM:**

import java.io.\*;

import java.util.\*;

public class EBbillCalculation

{

int cno;

String cname;

int pm\_reading,cm\_reading,units;

double billpay;

void getdetails()

{

Scanner in = new Scanner(System.in);

System.out.println("Enter the Consumer No:");

cno=in.nextInt();

System.out.println("Enter the Consumer Name:");

cname=in.next();

System.out.println("Enter the Previous Month Reading:");

pm\_reading=in.nextInt();

System.out.println("Enter the Current Month Reading:");

cm\_reading=in.nextInt();

units= cm\_reading-pm\_reading;

System.out.println("Choose the type of EB Connection

1.Domestic connection 2. Commercial connection");

int ch=in.nextInt();

switch(ch)

{

case 1:

Domestic\_Calculation(units);

break;

case 2: Commercial\_Calculation(units);

break;

}

}

void Domestic\_Calculation(int units)

{

billpay = 0;

if(units<=100)

{

billpay=units\*1.00;

}

else if(units>100 && units<=200)

{

billpay=100\*1.00+(units-100)\*2.50;

}

else if(units>200 && units<=500)

{

billpay=100\*1.00+200\*2.50+(units-200)\*4.00;

}

else if(units>500)

{

billpay =100\*1.00+200\*2.50+500\*4.00+(units-500)\*6.00;

}

show();

System.out.println("Amount to be Paid : " + billpay);

}

void Commercial\_Calculation(int units)

{

billpay = 0;

if(units<=100)

{

billpay=units\*2.00;

}

else if(units>100 && units<=200)

{

billpay=100\*2.00+(units-100)\*4.50;

}

else if(units>200 && units<=500)

{

billpay=100\*2.00+200\*4.50+(units-200)\*6.00;

}

else if(units>500)

{

billpay =100\*2.00+200\*4.50+500\*6.00+(units-500)\*7.00;

}

show();

System.out.println("Amount to be Paid : " + billpay);

}

public void show()

{

System.out.println("Customer Number : " + cno);

System.out.println("Customer Name : " + cname);

System.out.println("Units Consumed : " + units);

}

public static void main(String[] args)

{

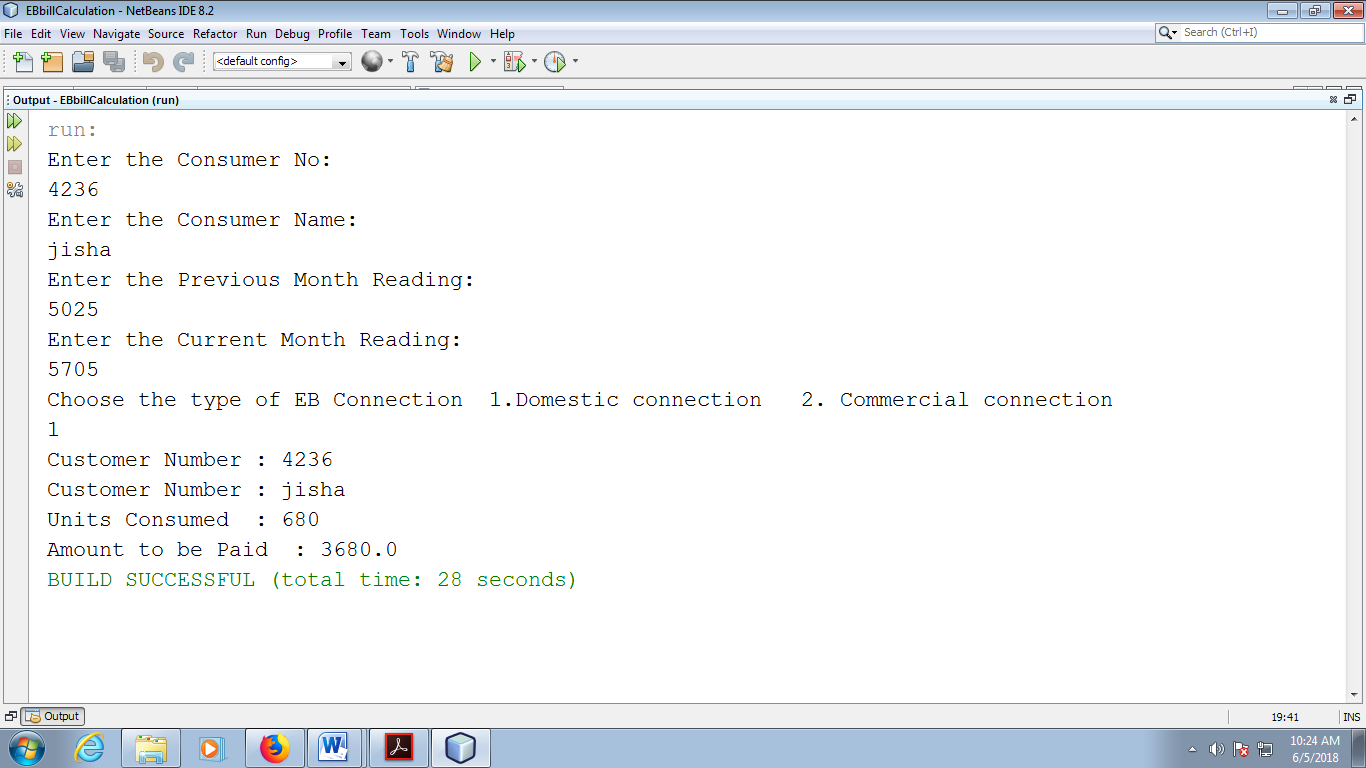
EBbillCalculation eb=new EBbillCalculation();

eb.getdetails();

}

}

**OUTPUT:**



**RESULT:**

Thus the java application for generating the electricity bill is created and executed successfully.

**EXPT: 2 a)**

**DATE:**

## APPLICATION USING PACKAGES

## CURRENCY CONVERTER

**AIM:**

To develop a java application to implement the currency converter (Dollar to INR, Euro to INR, Yen to INR and Vice Versa)

**ALGORITHM:**

Step 1: Start the netbeansIDE8.2

Step 2: Goto FILE🡪 new project🡪java🡪java application🡪click next

Step 3: Give the name for the application🡪Click Finish.

Step 4: Create a class called Currencyconverter takes the type of conversion from

user using switch case.

Step 5: Display the result and stop the program.

**PROGRAM:**

import java.io.\*;

import java.util.Scanner;

public class Currencyconverter {

public static void main(String[] args) {

double Rs;

Scanner in = new Scanner(System.in);

System.out.println("Choose the type of currency conversition");

System.out.println("1.USDollar to INDRupees,2.Euro to INDRupees,3. JapaneseYen to INDRupees");

int ch=in.nextInt();

switch(ch)

{

case 1: System.out.println("Dollar to Rupees conversion");

System.out.println("Enter the number of Dollars");

double dollar=in.nextInt();

Rs=dollar\*67.02;

System.out.println("USD="+dollar+" is INR="+Rs);

break;

case 2: System.out.println("EURO to Rupees conversion");

System.out.println("Enter the number of EURO");

double Euro=in.nextInt();

Rs=Euro\*78.29;

System.out.println("Euro="+Euro+" is INR="+Rs);

break;

case 3: System.out.println("Japanese Yen to Rupees conversion");

System.out.println("Enter the number of Yen");

double Yen=in.nextInt();

Rs=Yen\*0.61;

System.out.println("Yen="+Yen+" is INR="+Rs);

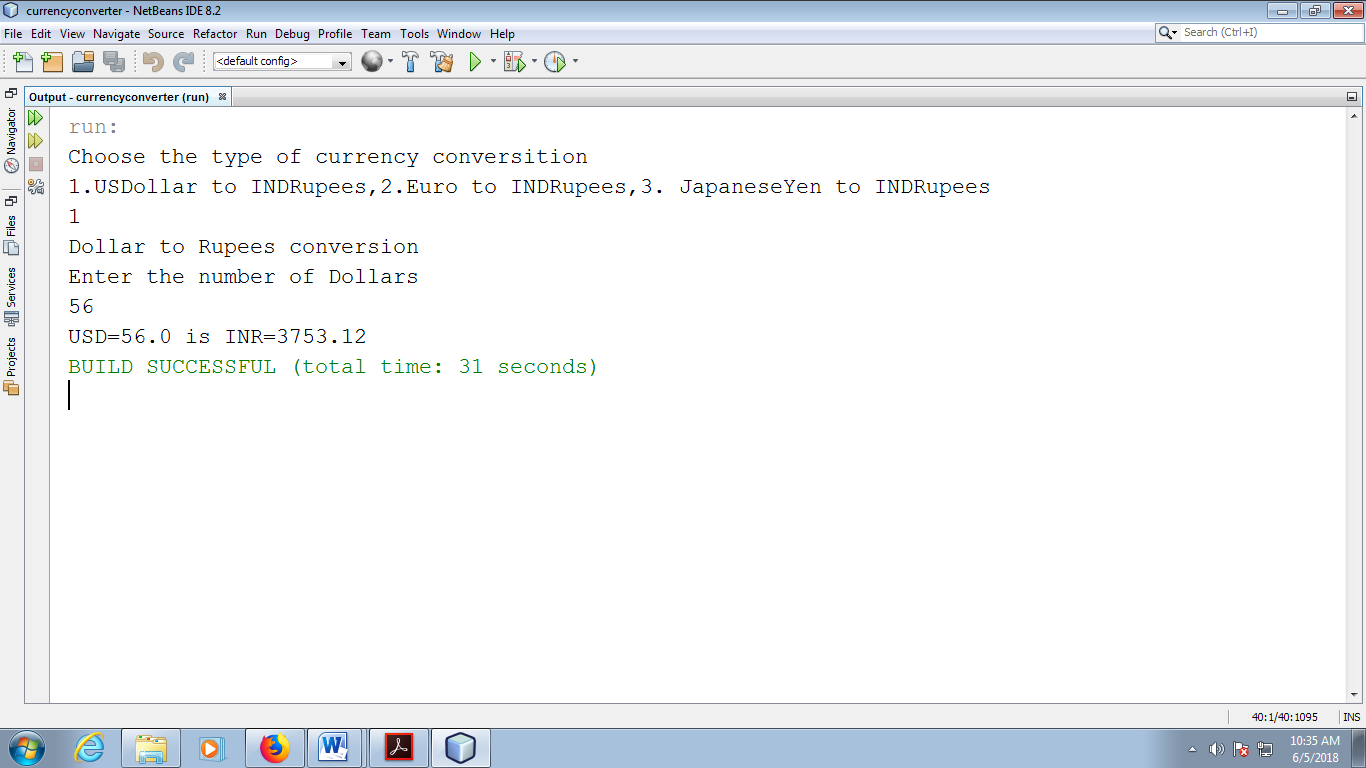
break;

}

}

}

**OUTPUT:**



**RESULT:**

Thus the java application to implement the currency converter is created and executed successfully.

**EXPT: 2 b)**

**DATE:**

## APPLICATION USING PACKAGES

## DISTANCE CONVERTER

**AIM:**

To develop a java application to implement the distance converter ( Meter to

Kilometer, Miles to Kilometers)

**ALGORITHM:**

Step 1: Start the netbeansIDE8.2

Step 2: Goto FILE🡪 new project🡪java🡪java application🡪click next

Step 3: Give the name for the application🡪Click Finish.

Step 4: Create a class called distanceconvertor takes the type of conversion from

user using switch case.

Step 5: Case1 chooses the Meters into KiloMeters and Case2 Miles to KiloMeters

Step 6: Display the result and stop the program.

**PROGRAM:**

import java.io.\*;

import java.util.Scanner;

public class distanceconvertor {

public static void main(String args[])

{

double meters,kilometers,miles;

Scanner in = new Scanner(System.in);

System.out.println("Choose the type of Distance conversion");

System.out.println("1.Meters to KiloMeters ,2. Miles to KiloMeters :");

int ch=in.nextInt();

switch(ch)

{

case 1: System.out.println("Meters into KiloMeters");

System.out.println("Enter the number of Meters");

meters=in.nextDouble();

kilometers = meters \* 0.001;

System.out.println( meters+ " Meters is " +kilometers+ " in Kilometers");

break;

case 2: System.out.println("Miles to Kilometers conversion");

System.out.println("Enter the number of Miles");

miles=in.nextDouble();

kilometers =miles \* 1.60934;

System.out.println( miles+ " Miles is " +kilometers + " in Kilometers");

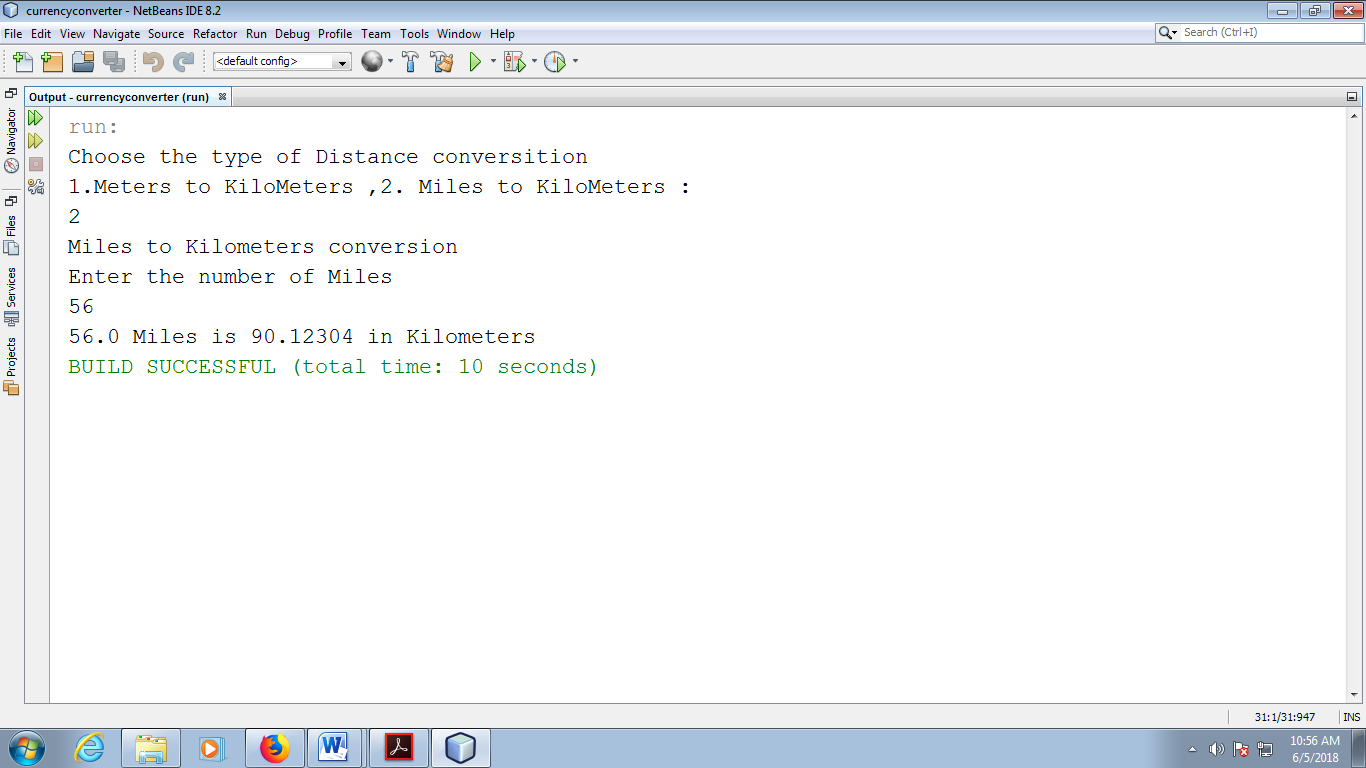
break;

}

}

}

**OUTPUT:**



**RESULT:**

Thus the java application to implement the distance converter is created and executed successfully.

**EXPT: 2 C)**

**DATE:**

## APPLICATION USING PACKAGES

## TIME CONVERTER

**AIM:**

To develop a java application to implement the time converter (hours to minutes, seconds and vice versa)

**ALGORITHM:**

Step 1: Start the netbeansIDE8.2

Step 2: Goto FILE🡪 new project🡪java🡪java application🡪click next

Step 3: Give the name for the application🡪Click Finish.

Step 4: Create a class called Timeconverter takes seconds as input form the

user and display the result to user.

Step 5: Stop the program.

**PROGRAM:**

import java.util.Scanner;

public class Timeconverter

{

public static void main(String[] args) {

int n,hr,min,sec;

Scanner in = new Scanner(System.in);

System.out.print("Enter the Input in seconds:");

n= in.nextInt();

if(n>3600){

min = n/60;

sec = n%60;

hr = min/60;

min = min%60;

System.out.println("Converted format:"+hr+ "hour " + min +"mins" + sec+"secs");

}

else{

min = n/60;

sec = n%60;

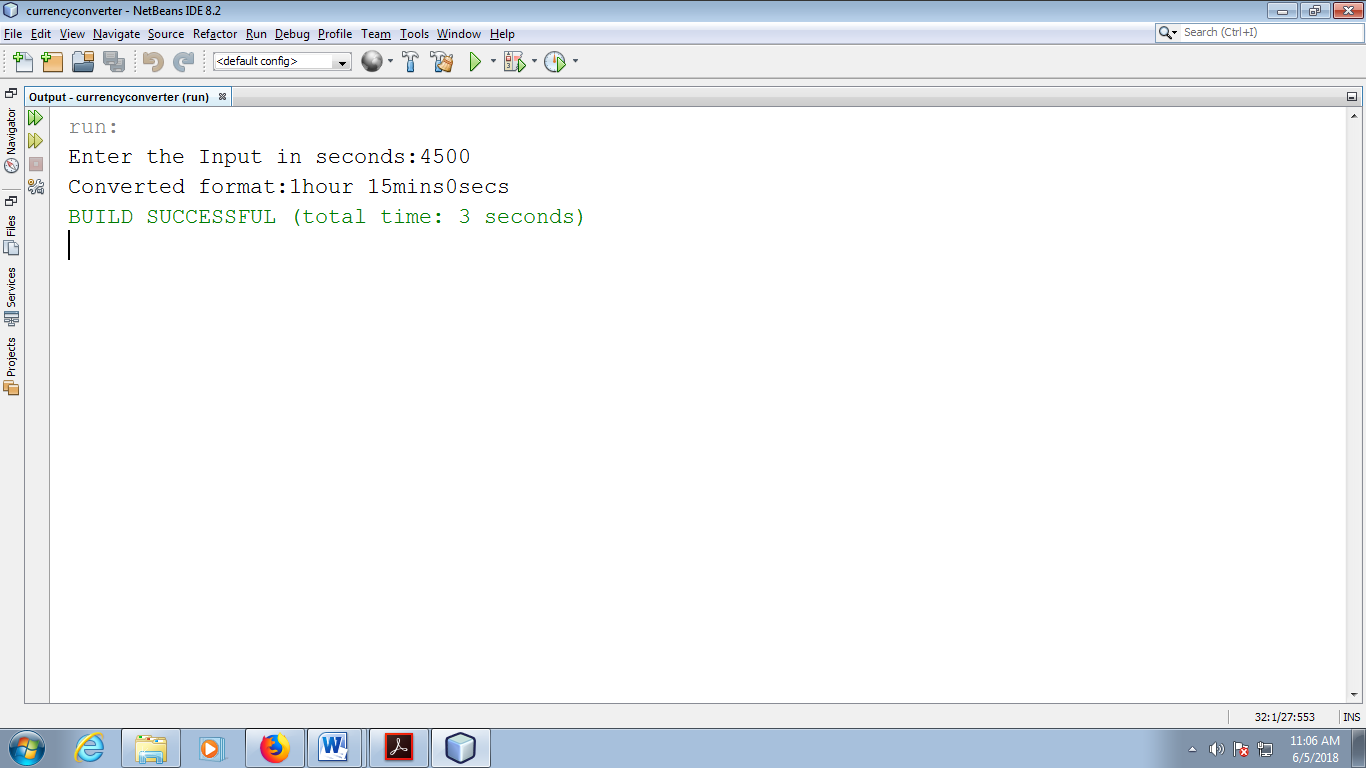
System.out.println("Converted format :"+min+" mins " +sec +"secs");

}

}

}

**OUTPUT:**



**RESULT:**

Thus the java application for time converter is created and executed successfully.

**EXPT: 3**

**DATE:**

**APPLICATION USING INHERITANCE**

**AIM:**

To develop a java application to calculate the payroll and generate the pay slip

for the employee

**ALGORITHM:**

Step 1: Start the netbeansIDE8.2

Step 2: Goto FILE🡪 new project🡪java🡪java application🡪click next

Step 3: Give the name for the application🡪Click Finish.

Step 4: Create a class called Emp and declare the necessary variables.

Step 5: Create the method getEmployeedetails() which collect the basic

information from the employee.

Step 6: Create the method pay\_calulation () for calculating the gross and net

salary from the basic pay of the employee.

Step 7: Print the output and stop execution.

**PROGRAM:**

import java.io.IOException;

import java.util.Scanner;

class Emp

{

String ename,Address,email;

int eid;

int mobile;

void getEmployeedetails()

{

Scanner in = new Scanner(System.in);

System.out.println("Enter the Emp\_id. :");

eid=in.nextInt();

System.out.println("Enter the Employee Name:");

ename=in.next();

System.out.println("Enter the Employee Address:");

Address=in.next();

System.out.println("Enter the Employee Email id :");

email=in.next();

System.out.println("Enter the Mobile No:");

mobile=in.nextInt();

}

void pay\_calulation(double BasicPay)

{

double DA,HRA,PF,Sfund,Gross\_Salary,Netsalary;

DA=BasicPay\*0.97;

HRA=BasicPay\*0.10;

PF=BasicPay\*0.12;

Sfund=BasicPay\*0.1;

Gross\_Salary=BasicPay+DA+HRA;

Netsalary=Gross\_Salary-(PF+Sfund);

System.out.println("Gross salary of the Employee"+Gross\_Salary);

System.out.println("Net salary of the Employee: "+Netsalary);

}

void display()

{

System.out.println("Emp\_id:"+eid);

System.out.println("Employee Name:"+ename);

System.out.println("Employee Address:"+Address);

System.out.println("Employee Email id :"+email);

System.out.println("Employee Mobile No:"+mobile);

}

}

class Programmer extends Emp

{

double BasicPay;

void Programmerdetails()

{

getEmployeedetails();

Scanner in = new Scanner(System.in);

System.out.println("Enter the Basic Pay of the Programmer:");

BasicPay=in.nextInt();

display();

pay\_calulation(BasicPay);

}

}

class AssistantProfessor extends Emp

{

void APDetails()

{

double BasicPay;

getEmployeedetails();

Scanner in = new Scanner(System.in);

System.out.println("Enter the Basic Pay of the AssistantProfessor:");

BasicPay=in.nextInt();

display();

pay\_calulation(BasicPay);

}

}

class AssociateProfessor extends Emp

{

double BasicPay;

void ASPDetails()

{

getEmployeedetails();

Scanner in = new Scanner(System.in);

System.out.println("Enter the Basic Pay of the AssociateProfessor:");

BasicPay=in.nextInt();

display();

pay\_calulation(BasicPay);

}

}

class Professor extends Emp

{

double BasicPay;

void profDetails()

{

getEmployeedetails();

Scanner in = new Scanner(System.in);

System.out.println("Enter the Basic Pay of the Professor:");

BasicPay=in.nextInt();

display();

pay\_calulation(BasicPay);

}

}

public class Employee

{

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

System.out.println("Choose the type Employee");

System.out.println("1.Programmer ,2.Assistant Professor,3.Associate Professor ,4.Professor: ");

int ch=in.nextInt();

switch(ch)

{

case 1: System.out.println("PROGRAMMER DETAILS");

Programmer p=new Programmer();

p.Programmerdetails();

break;

case 2: System.out.println("Assistant Professor DETAILS");

AssistantProfessor ap=new AssistantProfessor();

ap.APDetails();

break;

case 3: System.out.println("Associate Professor DETAILS");

AssociateProfessor asp=new AssociateProfessor();

asp.ASPDetails();

break;

case 4: System.out.println("Professor DETAILS");

Professor pf=new Professor();

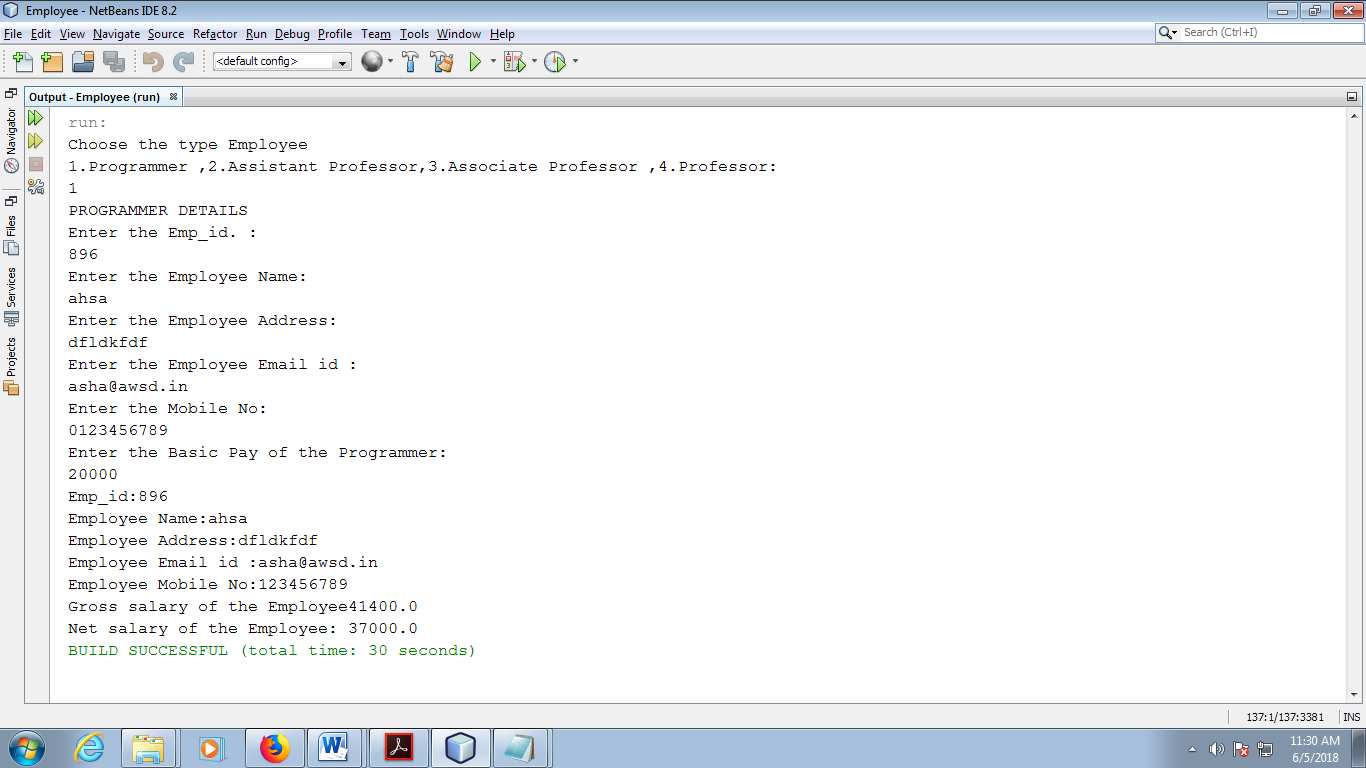
pf.profDetails();

break;

}

} }

**OUTPUT:**



**RESULT:**

Thus the java program for generating the employee pay slip is created and executed successfully.

**EXPT: 4**

**DATE:**

**ABSTRACT CLASS**

**AIM:**

To write a java program to create an abstract class named Shape.

**ALGORITHM:**

Step 1: Start the netbeansIDE8.2

Step 2: Goto FILE🡪 new project🡪java🡪java application🡪click next

Step 3: Give the name for the application🡪Click Finish.

Step 4: Create an abstract class named Shape that contains two integers and an

empty method named print Area()

Step 5: Create three classes named Rectangle, Triangle and Circle which

extends Shape

Step 6: Print the area of the given shapes.

Step 7: Stop the program.

**PROGRAM:**

package javaapplication3;

abstract class shape

{

int a=3,b=4;

abstract public void print\_area();

}

class rectangle extends shape

{

public int area\_rect;

@Override

public void print\_area()

{

area\_rect=a\*b;

System.out.println("The area of rectangle is:"+area\_rect);

}

}

class triangle extends shape

{

int area\_tri;

@Override

public void print\_area()

{

area\_tri=(int) (0.5\*a\*b);

System.out.println("The area of triangle is:"+area\_tri);

}

}

class circle extends shape

{

int area\_circle;

@Override

public void print\_area()

{

area\_circle=(int) (3.14\*a\*a);

System.out.println("The area of circle is:"+area\_circle);

}

}

public class JavaApplication3 {

public static void main(String[] args) {

rectangle r=new rectangle();

r.print\_area();

triangle t=new triangle();

t.print\_area();

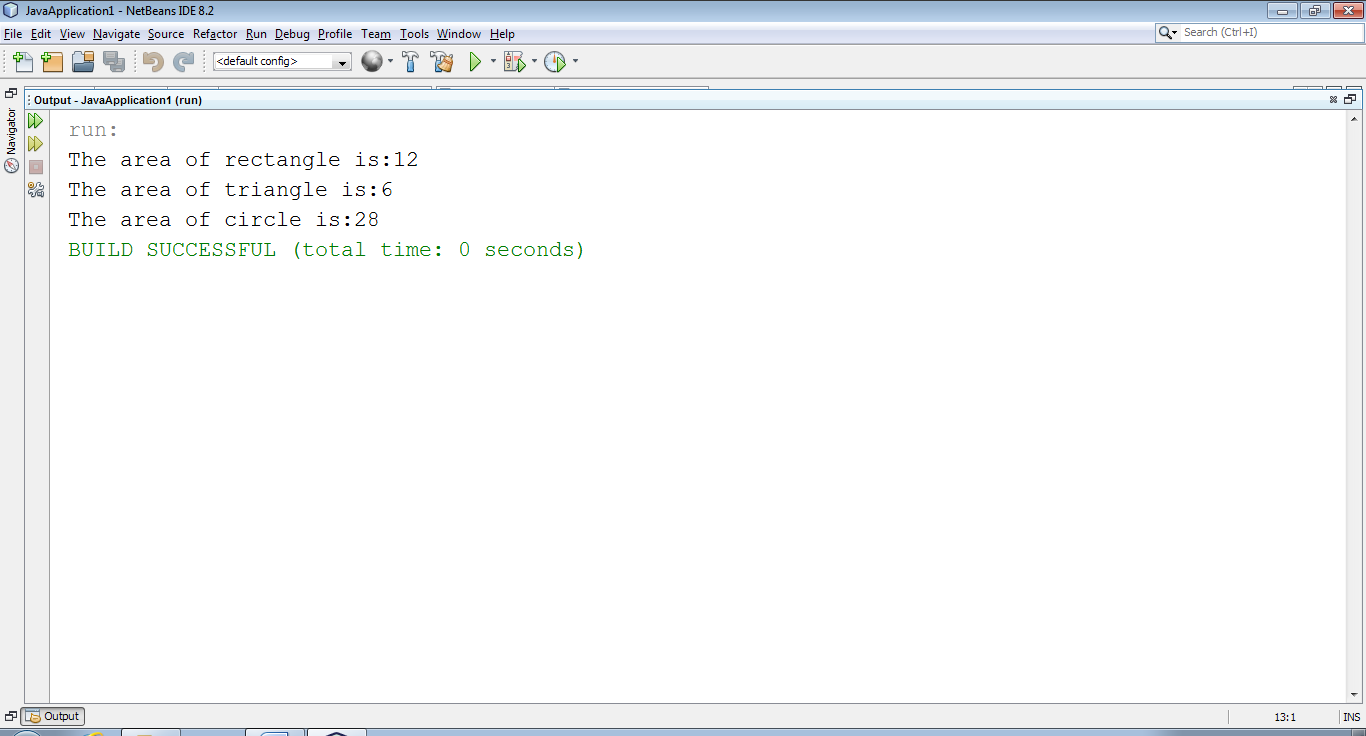
circle r1=new circle();

r1.print\_area();

}

}

**OUTPUT:**



**RESULT:**

Thus the program for abstract class has been written and executed successfully.

**EXPT: 5**

**DATE:**

**MULTIPLE INHERITANCE USING INTERFACAE**

**AIM:**

To write a java program that demonstrates Multiple Inheritance using Interface.

**ALGORITHM:**

Step 1: Start the netbeansIDE8.2

Step 2: Goto FILE🡪 new project🡪java🡪java application🡪click next

Step 3: Give the name for the application🡪Click Finish.

Step 4: Define two interfaces:

    → interface A with method meth1()

    → interface B with method meth2()

Step 5: Create a class MyClass that implements both interfaces A and B.

Step 6: Provide concrete implementations for meth1() and meth2() inside

MyClass.

Step 7: In the main() method of the MultipleInheritanceDemo class:

    a. Create an object of MyClass.

    b. Call meth1() and meth2() using the object.

Step 8: Observe the output printed from both method implementations.

Step 9: Stop the program.

**PROGRAM:**

// First interface

interface A {

void meth1();

}

// Second interface

interface B {

void meth2();

}

// Class implementing both A and B

class MyClass implements A, B {

public void meth1() {

System.out.println("Implement meth1() from interface A.");

}

public void meth2() {

System.out.println("Implement meth2() from interface B.");

}

}

// Test class

public class MultipleInheritanceDemo {

public static void main(String[] args) {

MyClass obj = new MyClass();

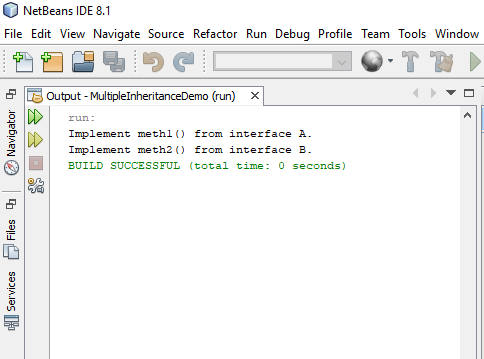
obj.meth1();

obj.meth2();

}

}

**OUTPUT:**



**RESULT:**

Thus the program to demonstrate multiple inheritance using interface has been written and executed successfully.

**EXPT: 6**

**DATE:**

**STRING AND STRINGBUFFER CLASS**

**AIM:**

To write a java program to perform string operations using String and StringBuffer class.

**ALGORITHM:**

Step 1: Start the netbeansIDE8.2

Step 2: Goto FILE🡪 new project🡪java🡪java application🡪click next

Step 3: Give the name for the application (e.g., StringOperationsDemo)🡪Click

Finish.

Step 4: In the newly created project, open the main class file (e.g.,

StringOperationsDemo.java).

Step 5: Inside the main() method:

* Declare two String variables.
* Perform operations like concatenation, length, charAt, substring, toUpperCase, toLowerCase, indexOf using String class methods.

Step 6: Create a StringBuffer object.

* Perform append, insert, replace, delete, reverse operations using StringBuffer class methods.
* Print the output after each operation.

Step 7: Stop the program.

**PROGRAM:**

public class StringOperationsDemo {

public static void main(String[] args) {

// --- Using String class (Immutable) ---

String str1 = "Java";

String str2 = "Programming";

// Concatenation

String result = str1 + " " + str2;

System.out.println("Concatenated String: " + result);

// Length

System.out.println("Length of result string: " + result.length());

// Character at position

System.out.println("Character at index 5: " + result.charAt(5));

// Substring

System.out.println("Substring from index 5 to 11: " + result.substring(5, 11));

// To Uppercase

System.out.println("Uppercase: " + result.toUpperCase());

// To Lowercase

System.out.println("Lowercase: " + result.toLowerCase());

// Index of a character

System.out.println("Index of 'g': " + result.indexOf('g'));

// --- Using StringBuffer class (Mutable) ---

StringBuffer sb = new StringBuffer("Hello");

// Append

sb.append(" World");

System.out.println("\nAfter append: " + sb);

// Insert

sb.insert(6, "Java ");

System.out.println("After insert: " + sb);

// Replace

sb.replace(6, 10, "Awesome");

System.out.println("After replace: " + sb);

// Delete

sb.delete(6, 13);

System.out.println("After delete: " + sb);

// Reverse

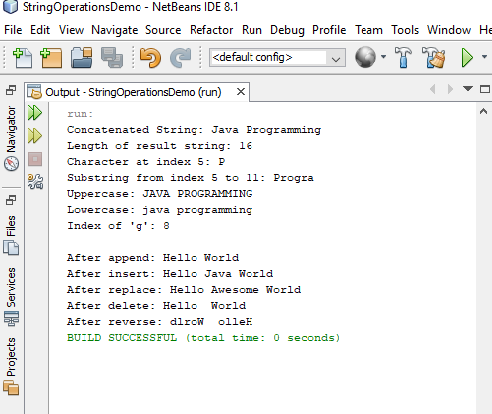
sb.reverse();

System.out.println("After reverse: " + sb);

}

}

**OUTPUT:**



**RESULT:**

Thus the program to perform string operations using String and StringBuffer classs has been written and executed successfully.

**EXPT: 7**

**DATE:**

# USER DEFINED EXCEPTION HANDLING

**AIM:**

To write a Java program to implement user defined exception handling.

**ALGORITHM:**

Step 1: Start the netbeansIDE8.2

Step 2: Goto FILE🡪 new project🡪java🡪java application🡪click next

Step 3: Give the name for the application🡪Click Finish.

Step 4: Create a user defined exception called MyException which extends class

Exception.

Step 5: Create try-catch block to handle the exception.

Step 6: Throw an exception of user defined type as an argument in main()

Step 7: Exception is handled using try, catch block

Step 8: Display the user defined exception.

**PROGRAM:**

package example1;

class MyException extends Exception{

String str1;

MyException(String str2) {

str1=str2;

}

public String toString(){

return ("MyException Occurred: "+str1) ;

}

}

public class Example1 {

public static void main(String[] args)

{

try{

System.out.println("Starting of try block");

// I'm throwing the custom exception using throw

throw new MyException("This is My error Message");

}

catch(MyException exp){

System.out.println("Catch Block") ;

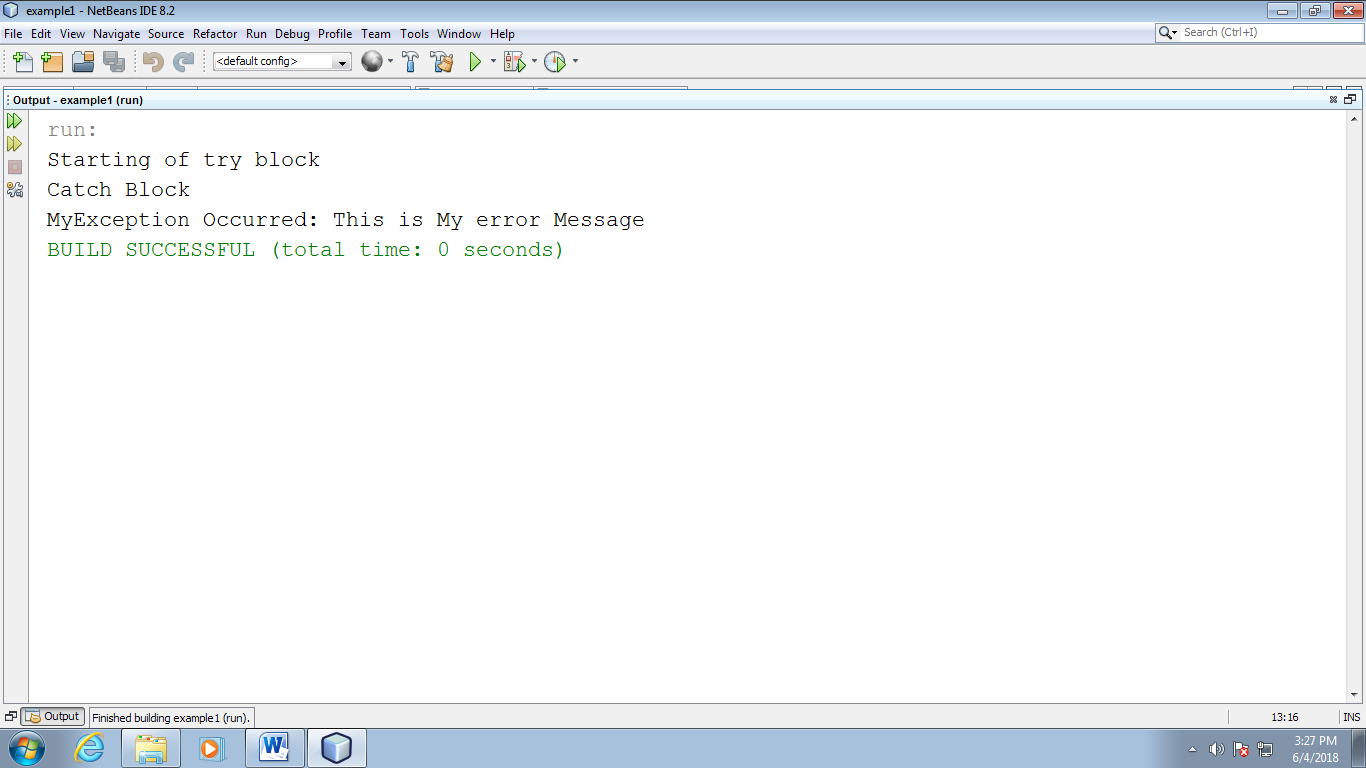
System.out.println(exp) ;

}

}

}

**OUTPUT:**



**RESULT:**

Thus the program user defined exception handling has been written and executed successfully.

**EXPT: 8**

**DATE:**

# FILE OPERATIONS

**AIM:**

To write a Java program to implement the concepts of file.

**ALGORITHM:**

Step 1: Start the program

Step 2: Open the notepad and type the program and save the program as

Filedemo.java

Step 3: Click on the Cmd Prompt 🡪 cd C:\Java\jdk1.8\bin

Step 4: Now compile the Filedemo.java using

javac Filedemo.java

Step 5: Before executing the Filedemo.java create another java file using notepad

called Fib.java

Step 6: Execute the program using

java Filedemo.java and give the input file name Fib.java

Step 7: Stop the program.

**PROGRAM:**

import java.util.Scanner;

import java.io.File;

public class Filedemo {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

String s=input.nextLine();

File f1=new File(s);

System.out.println("File Name:"+f1.getName());

System.out.println("Path:"+f1.getPath());

System.out.println("Abs Path:"+f1.getAbsolutePath());

System.out.println("Parent:"+f1.getParent());

System.out.println("This file is:"+(f1.exists()?"Exists":"Does not exists"));

System.out.println("Is file:"+f1.isFile());

System.out.println("Is Directory:"+f1.isDirectory());

System.out.println("Is Readable:"+f1.canRead());

System.out.println("IS Writable:"+f1.canWrite());

System.out.println("Is Absolute:"+f1.isAbsolute());

System.out.println("File Last Modified:"+f1.lastModified());

System.out.println("File Size:"+f1.length()+"bytes");

System.out.println("Is Hidden:"+f1.isHidden());

}

}

**OUTPUT:**

**Fib.java**

public class Fib {

public static void main(String[] args) {

int n1=0,n2=1,n3,i,count=10;

System.out.print(n1+" "+n2);

for(i=2;i<count;++i)

{

n3=n1+n2;

System.out.print(" "+n3);

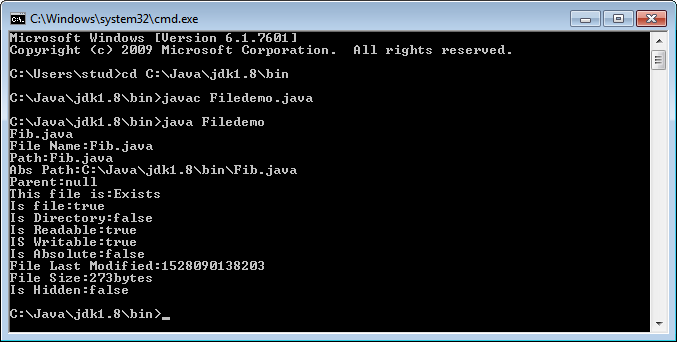
n1=n2;

n2=n3;

}

}

}



**RESULT:**

Thus the program for implementing the file concept has been written and executed successfully.

**EXPT: 9**

**DATE:**

# MULTI THREADING

**AIM:**

To write a Java program to implement the concepts of multithreading.

**ALGORITHM:**

Step 1: Start the netbeansIDE8.2

Step 2: Goto FILE🡪 new project🡪java🡪java application🡪click next

Step 3: Give the name for the application🡪Click Finish.

Step 4: Create three threads A, odd, even

Step 5: Thread A generates the random number

Step 6: If random number is even ,Thread even will display the square of the

random number

Step 7: If random number is odd ,Thread odd will display the cube of the random

number

**PROGRAM:**

package mtherad;

import java.util.\*;

class even implements Runnable{

public int x;

public even(int x){

this.x=x;

}

@Override

public void run()

{

System.out.println("Thread Name:Even Thread and square is: " + x \* x);

}

}

class odd implements Runnable{

public int x;

public odd(int x){

this.x=x;

}

@Override

public void run()

{

System.out.println("Thread Name:Odd Thread and cube is :"+ x \* x \* x);

}

}

class A extends Thread{

public String tname;

public Random r;

public Thread t1,t2;

public A(String s){

tname=s;

}

@Override

public void run()

{

int num=0;

r=new Random();

try {

for(int i=0;i<50;i++){

num=r.nextInt(100);

System.out.println("main thread and generated number is"+num);

if(num%2==0)

{

t1=new Thread(new even(num));

t1.start();

}else{

t2=new Thread(new odd(num));

t2.start();

}

Thread.sleep(1000);

System.out.println("------------------------------------");

}

}

catch(InterruptedException ex)

{

System.out.println(ex.getMessage());

}

}

}

public class Mtherad {

public static void main(String[] args) {

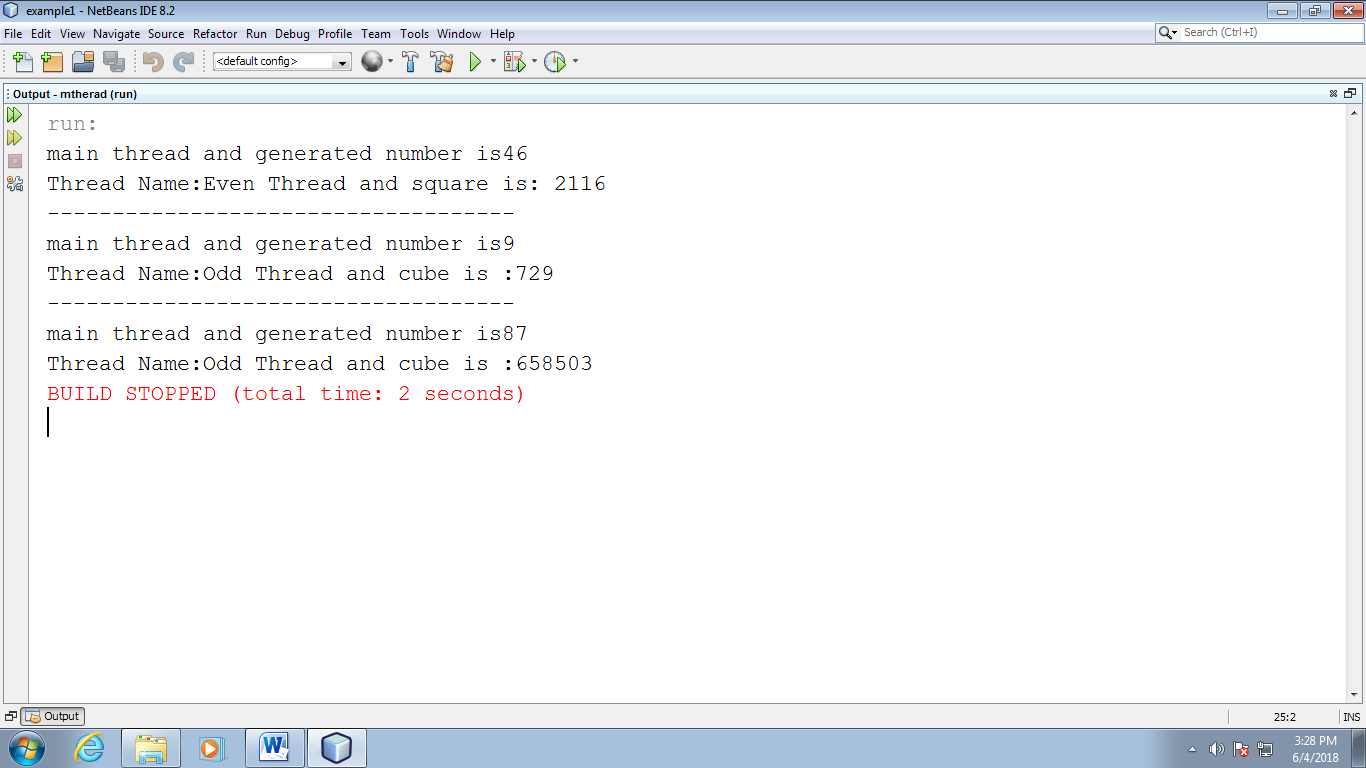
A a=new A("one");

a.start();

}

}

**OUTPUT:**



**RESULT:**

Thus the program for implementing the concept of multithreading has been written and executed successfully.

**EXPT: 10**

**DATE:**

# COLLECTION FRAMEWORK

**AIM:**

To write a Java program to demonstrate the use of core Collection Framework classes like ArrayList, HashSet, and HashMap.

**ALGORITHM:**

Step 1: Start the netbeansIDE8.2

Step 2: Goto FILE🡪 new project🡪java🡪java application🡪click next

Step 3: Give the name for the application (CollectionFrameworkDemo)🡪Click

Finish.

Step 4: Inside the main() method:

* Create an ArrayList, add elements, print the list.
* Create a HashSet, add elements including duplicates, print the set.
* Create a HashMap, insert key-value pairs, update one value, print using a loop.

Step 5: Save and Run the program (Shift + F6).

Step 6: Observe how each collection behaves differently.

Step 6: Stop the program.

**PROGRAM:**

import java.util.\*;

public class CollectionFrameworkDemo {

public static void main(String[] args) {

// ----- ArrayList (List interface - Ordered, allows duplicates) -----

System.out.println("---- ArrayList Demo ----");

ArrayList<String> list = new ArrayList<>();

list.add("Apple");

list.add("Banana");

list.add("Cherry");

list.add("Apple"); // Duplicate allowed

System.out.println("ArrayList: " + list);

// ----- HashSet (Set interface - Unordered, no duplicates) -----

System.out.println("\n---- HashSet Demo ----");

HashSet<String> set = new HashSet<>();

set.add("Red");

set.add("Green");

set.add("Blue");

set.add("Red"); // Duplicate ignored

System.out.println("HashSet: " + set);

// ----- HashMap (Map interface - Key-Value pairs) -----

System.out.println("\n---- HashMap Demo ----");

HashMap<Integer, String> map = new HashMap<>();

map.put(101, "John");

map.put(102, "Alice");

map.put(103, "Bob");

map.put(102, "Emma"); // Overwrites value for key 102

System.out.println("HashMap: " + map);

// Iterate over HashMap

System.out.println("\nIterating over HashMap:");

for (Map.Entry<Integer, String> entry : map.entrySet()) {

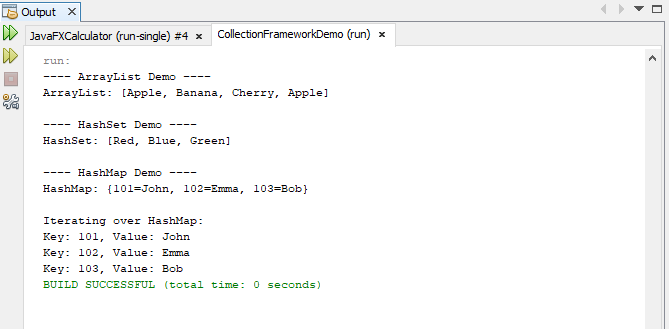
System.out.println("Key: " + entry.getKey() + ", Value: " + entry.getValue());

}

}

}

**OUTPUT:**



**RESULT:**

Thus the program to demonstrate the use of core Collection Framework classes like ArrayList, HashSet, and HashMap has been written and executed successfully.

**EXPT: 11**

**DATE:**

# CALCULATOR

**AIM:**

To write a Java program to develop GUI-based Calculator applications using JavaFX controls, layouts and menus.

**ALGORITHM:**

Step 1: Start the program

Step 2: Open the notepad and type the program and save the program as

Calculator.java

Step 3:Create the class Calculator. Define and declare its variables.

Step 4: Using Calculator constructor create bottons.

Step 5: Using actionPerformed() method define the function that has to be done

when the corresponding button is pressed.

Step 6: Click on the Cmd Prompt 🡪 cd C:\Java\jdk1.8\bin

Step 7: Now compile the Filedemo.java using javac Calculator.java

Step 8: Execute the program using command java Calculator

**PROGRAM:**

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.\*;

import javafx.stage.Stage;

import javafx.geometry.\*;

import javafx.event.ActionEvent;

import javafx.scene.input.KeyCode;

import javafx.scene.input.KeyEvent;

public class JavaFXCalculator extends Application {

private TextField display;

private boolean startNewNumber = true;

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage primaryStage) {

display = new TextField();

display.setEditable(false);

display.setAlignment(Pos.CENTER\_RIGHT);

display.setPrefHeight(50);

GridPane grid = new GridPane();

grid.setHgap(10);

grid.setVgap(10);

grid.setPadding(new Insets(10));

String[][] buttons = {

{"C", "", "", "", ""},

{"7", "8", "9", "/", ""},

{"4", "5", "6", "\*", ""},

{"1", "2", "3", "-", ""},

{"0", ".", "=", "+", ""}

};

for (int row = 0; row < buttons.length; row++) {

for (int col = 0; col < buttons[row].length; col++) {

String label = buttons[row][col];

if (!label.isEmpty()) {

Button btn = new Button(label);

btn.setPrefSize(50, 50);

btn.setOnAction(this::processInput);

grid.add(btn, col, row);

}

}

}

VBox root = new VBox(10, display, grid);

root.setPadding(new Insets(10));

Scene scene = new Scene(root, 310, 320);

primaryStage.setTitle("JavaFX Calculator");

primaryStage.setScene(scene);

primaryStage.setResizable(false);

primaryStage.show();

scene.setOnKeyPressed(this::handleKey);

}

private void processInput(ActionEvent e) {

String value = ((Button) e.getSource()).getText();

if (value.equals("=")) {

calculate();

} else if (value.equals("C")) {

display.clear();

startNewNumber = true;

} else if ("+-\*/".contains(value)) {

handleInput(" " + value + " ");

} else {

handleInput(value);

}

}

private void handleInput(String value) {

if (startNewNumber) {

display.clear();

startNewNumber = false;

}

display.appendText(value);

}

private void calculate() {

try {

String[] tokens = display.getText().trim().split(" ");

if (tokens.length < 3) return;

double result = Double.parseDouble(tokens[0]);

for (int i = 1; i < tokens.length; i += 2) {

String op = tokens[i];

double next = Double.parseDouble(tokens[i + 1]);

switch (op) {

case "+":

result += next;

break;

case "-":

result -= next;

break;

case "\*":

result \*= next;

break;

case "/":

if (next == 0) {

display.setText("Error: Divide by 0");

startNewNumber = true;

return;

}

result /= next;

break;

default:

display.setText("Error");

return;

}

}

display.setText(String.valueOf(result));

startNewNumber = true;

} catch (Exception ex) {

display.setText("Syntax Error");

startNewNumber = true;

}

}

private void handleKey(KeyEvent event) {

String key = event.getText();

KeyCode code = event.getCode();

if ("0123456789.".contains(key)) {

handleInput(key);

} else if ("+-\*/".contains(key)) {

handleInput(" " + key + " ");

} else if (code == KeyCode.ENTER || key.equals("=")) {

calculate();

} else if (code == KeyCode.DELETE) {

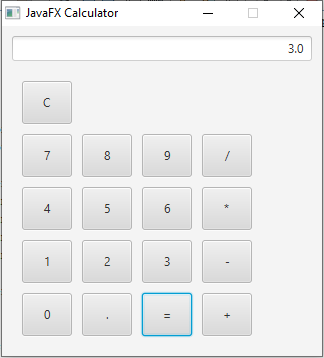
display.clear();

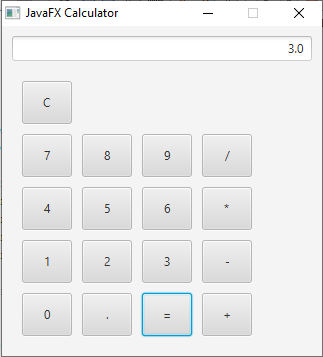
}

}

}

**OUTPUT:**





**RESULT:**

Thus the Java program to develop GUI-based Calculator applications using JavaFX controls, layouts and menus has been written and executed successfully.

**EXPT: 12**

**DATE:**

# STUDENT DATABASE MANAGEMENT SYSTEM

### OBJECTIVE:

It provides a common platform to connect student and teacher online. The registered teacher can create Quiz and student can take quiz and can assess himself/herself.

### Users of the System

1. Teacher
2. Student

### Functional Requirements

**1. Teacher**

1. Can create quiz after getting logged in!
2. Can enter subjects and enter question with its options and answer at the time of creating quiz.
3. 10 Question for each quiz required to be completed.

**2. Student**

1. Can search quiz according to their interest.
2. Click on the id of quiz and ready to start it just clicking on a button.
3. After completing all questions, result will be displayed Automatically.
4. Can view the description about each and every question in the respective quiz.

### Non-Functional Requirements

1. Secure access of confidential data (user’s details). SSL can be used.
2. 24 X 7 availability
3. Browser testing and support for IE, NN, Mozilla, and Firefox
4. Reports exportable in .XLS, .PDF
5. Create a detailed UML diagram (Component, Sequence, Class) for the system and its sub-components

### User Interface Priorities

1. Professional look and feel
2. Use of AJAX at least with all registration forms and with every search option and at the id of each searched result with on mouse over event.

### Tools to be used

1. Use any IDE to develop the project. It may be My eclipse / Eclipse / Netbeanse.
2. Oracle 10g for the database.
3. Server: Apache Tomcat/JBoss/Glassfish/Web logic/Websphere.

### Front End and Back End

1. **Front End:** JSP, JDBC, Javascript, AJAX
2. **Back End:** Oracle

**PROCEDURE:**

**Step 1:** Start the Net beans IDE 8.1 . choose File🡪 new project 🡪java web🡪java web application

**Step 2:** Click🡪next🡪give the project name🡪next.

**Step 3:** Choose Hibernate 4.3.1**🡪**click Finish.

**Step 4:** Edit the contents in index. Html and right click project🡪 new 🡪 servlet

**Step 5:** Give the class name as quizservlet 🡪 finish

**Step 6:** Choose the All programs🡪MySQL🡪MySQL Server 5.7🡪 MySQL Command Client

**Step 7:** Enter the password as root

**Step 8:** Create a database using the command

**mysql > CREATE DATABASE veena;**

**Step 9:** Create a table in the database using the command

**mysql > CREATE TABLE veena.student(seatno int(5),name varchar(20),**

**total int(5));**

**Step 10:** Right click the libraries🡪Add jar/folder🡪 C:\Program Files\MySQL🡪 choose **mysql-connectot-java**

**Step 12:** Compile the program and execute it.

**Step 13:** Stop the program.

**PROGRAM**

**SERVLET CODE:**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.\*;

import java.sql.\*;

import java.net.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

/\*\*

\*

\* @author stud

\*/

@WebServlet(urlPatterns = {"/quizservlet"})

public class quizservlet extends HttpServlet {

String msg,seatno,name,ans1,ans2,ans3,ans4,ans5,ans6,ans7,ans8,ans9,ans10;

int total=0;

Connection connect;

Statement stmt =null;

ResultSet rs=null;

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException

{

try

{

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

connect =DriverManager.getConnection("jdbc:mysql://localhost:3306/veena","root","root");

msg="Thank you";

}

catch(ClassNotFoundException cnfex)

{

cnfex.printStackTrace();

}

catch(SQLException e)

{

e.printStackTrace();

}

seatno=request.getParameter("seatno");

name=request.getParameter("name");

ans1=request.getParameter("r0");

ans2=request.getParameter("r1");

ans3=request.getParameter("r2");

ans4=request.getParameter("r3");

ans5=request.getParameter("r4");

ans6=request.getParameter("r5");

ans7=request.getParameter("r6");

ans8=request.getParameter("r7");

ans9=request.getParameter("r8");

ans10=request.getParameter("r9");

if(ans1.equals("four"))

total+=2;

if(ans2.equals("four"))

total+=2;

if(ans3.equals("four"))

total+=2;

if(ans4.equals("four"))

total+=2;

if(ans5.equals("four"))

total+=2;

if(ans6.equals("four"))

total+=2;

if(ans7.equals("four"))

total+=2;

if(ans8.equals("four"))

total+=2;

if(ans9.equals("four"))

total+=2;

if(ans10.equals("four"))

total+=2;

try

{

Statement stmt=connect.createStatement();

String query="INSERT INTO veena.student("+"seatno,name,total"+")VALUES('"+seatno+"','"+name+"','"+total+"')";

int result=stmt.executeUpdate(query);

stmt.close();

}

catch(SQLException e1)

{

}

response.setContentType("text/html");

PrintWriter out =response.getWriter();

out.println("<html><body><center>");

out.println("<h1>"+msg+"</h1>\n");

out.println("<h3>Your result is "+"</h3></br>");

out.println("<table border=2>");

try

{

Statement stmt =connect.createStatement();

String query ="SELECT \* FROM student";

rs=stmt.executeQuery(query);

out.println("<th>"+"seatno"+"</th>");

out.println("<th>"+"name"+"</th>");

out.println("<th>"+"marks"+"</th>");

while(rs.next())

{

out.println("<tr>");

out.println("<td>"+rs.getInt(1)+"</td>");

out.println("<td>"+rs.getString(2)+"</td>");

out.println("<td>"+rs.getString(3)+"</td>");

out.println("</tr>");

}

out.println("</table>");

}

catch(SQLException ex)

{

}

finally

{ try

{

if(rs!=null)

rs.close();

if(stmt!=null)

stmt.close();

if(connect!=null)

connect.close();

}

catch(SQLException e)

{

}

}

out.println("</center>");

out.println("</body></html>");

}

}

**HTML CODE:**

<html>

<head>

<title>Quiz></title>

</head>

<body>

<h2>Hello , Welcome to Quiz!</h2>

<form name="frm" action="quizservlet" method="post">

Seat no:<input type="text" name="seatno" />

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

<h2>Hello , Welcome to Quiz!</h2>

<p>1.What is HTML?</p><br />

<input type="radio" name="r0" value="one"/>Highlevel makeup language<br/>

<input type="radio" name="r0" value="two"/>HyperText Makeup language<br/>

<input type="radio" name="r0" value="three"/>Highlighter markup language<br/>

<input type="radio" name="r0" value="four"/>Hypertext markup language<br/>

<p>2.What is XML</p><br />

<input type="radio" name="r1" value="one"/>Extendable Markup Language<br/>

<input type="radio" name="r1" value="two"/>Extensible Makeup Language<br/>

<input type="radio" name="r1" value="three"/> Extension of Markup Language<br/>

<input type="radio" name="r1" value="four"/>Extensible Markup Language<br/>

<p>3.What is DOM?</p><br />

<input type="radio" name="r2" value="one"/>Domain Object Model <br/>

<input type="radio" name="r2" value="two"/>Domains Object Model<br/>

<input type="radio" name="r2" value="three"/>Domain Of Model <br/>

<input type="radio" name="r2" value="four"/>Document Object Model<br/>

<p>4.What is DHTML ?</p><br />

<input type="radio" name="r3" value="one"/>Dynamic of HTML<br/>

<input type="radio" name="r3" value="two"/>Digital HTML<br/>

<input type="radio" name="r3" value="three"/>Direct Link<br/>

<input type="radio" name="r3" value="four"/>Dynamic HTML<br/>

<p>5.What is the name of code used for color</p><br />

<input type="radio" name="r4" value="one"/> MMM<br/>

<input type="radio" name="r4" value="two"/>GRB<br/>

<input type="radio" name="r4" value="three"/>DFR <br/>

<input type="radio" name="r4" value="four"/>RGB<br/>

<p>6.What is INFITT </p><br />

<input type="radio" name="r5" value="one"/>International Forum For Information Technology <br/>

<input type="radio" name="r5" value="two"/>International Forum For Information Technology in Telugu<br/>

<input type="radio" name="r5" value="three"/>International Forum For Information Transfer in Tamil <br/>

<input type="radio" name="r5" value="four"/>International Forum For Information Technology in Tamil<br/>

<p>7.The Additional Info given to a tag is </p><br />

<input type="radio" name="r6" value="one"/>Objects <br/>

<input type="radio" name="r6" value="two"/>Container<br/>

<input type="radio" name="r6" value="three"/>Sub Tag <br/>

<input type="radio" name="r6" value="four"/>Attributes<br/>

<p>8.Loop can be used with which tag ?</p><br />

<input type="radio" name="r7" value="one"/>input<br/>

<input type="radio" name="r7" value="two"/>img<br/>

<input type="radio" name="r7" value="three"/>bgcolor<br/>

<input type="radio" name="r7" value="four"/>bgsound<br/>

<p>9.Tag for creating line is </p><br />

<input type="radio" name="r8" value="one"/>lb<br/>

<input type="radio" name="r8" value="two"/>line<br/>

<input type="radio" name="r8" value="three"/>br<br/>

<input type="radio" name="r8" value="four"/>hr<br/>

<p>10.face attibute is for which tag</p><br />

<input type="radio" name="r9" value="one"/>bgsound<br/>

<input type="radio" name="r9" value="two"/>img<br/>

<input type="radio" name="r9" value="three"/>marquee <br/>

<input type="radio" name="r9" value="four"/>font<br/>

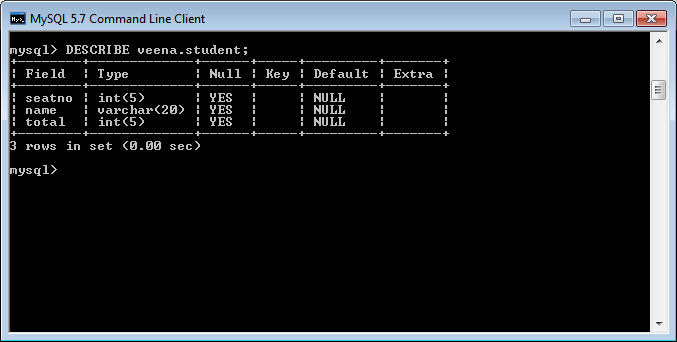
<input type="submit" value="submit answer" >

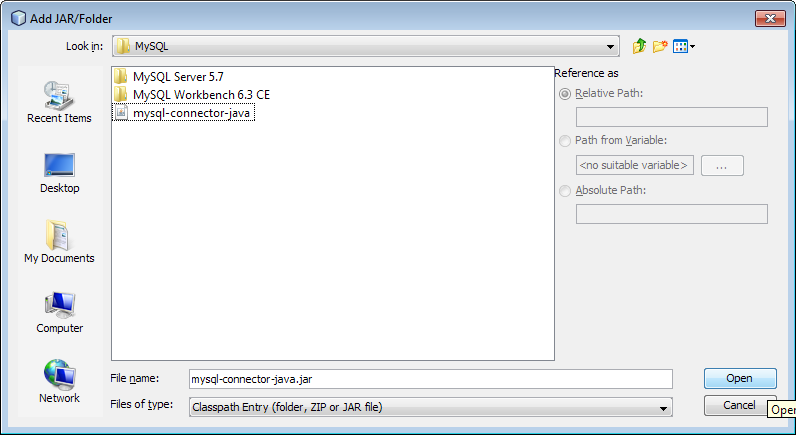
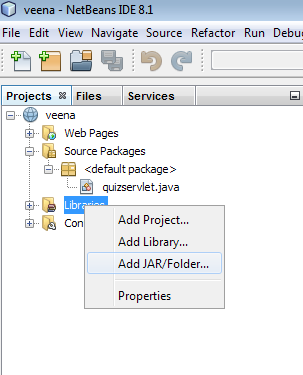
</form>

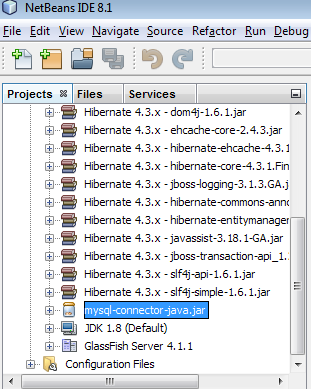
</body>

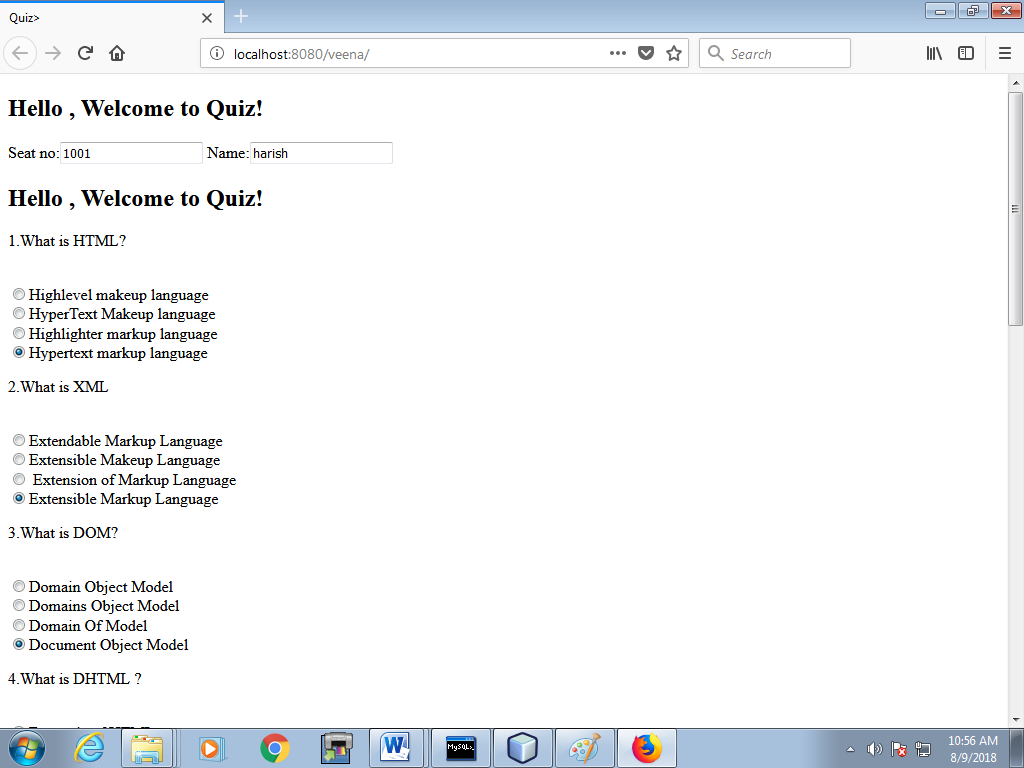
</html>

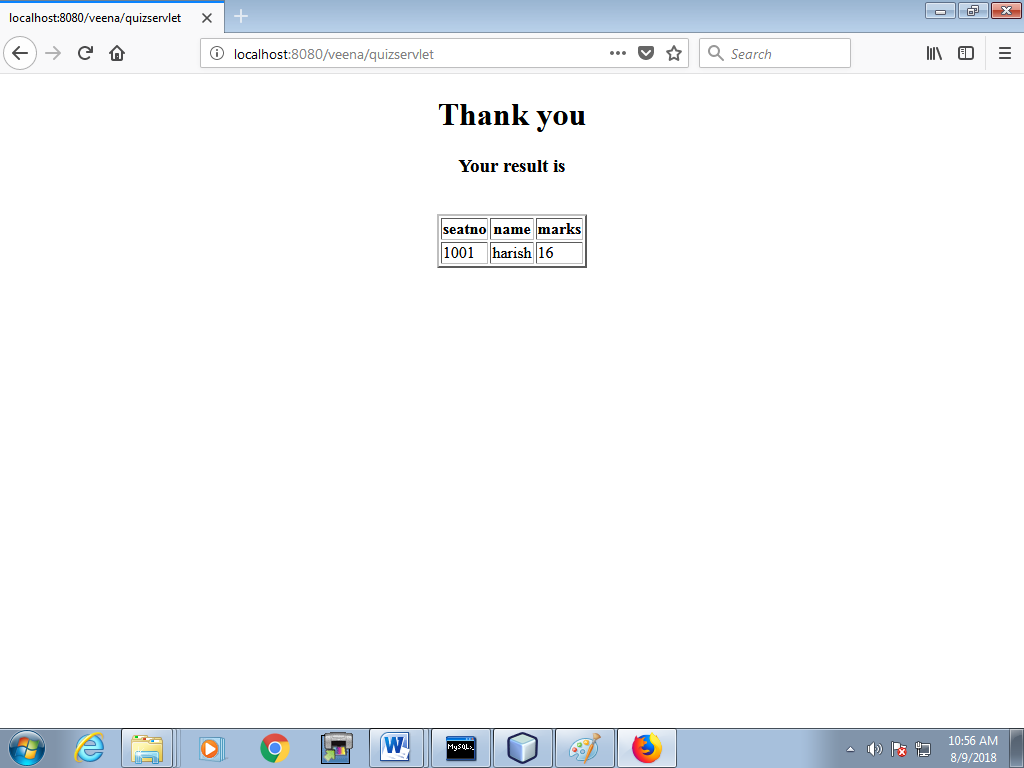
**OUTPUT:**

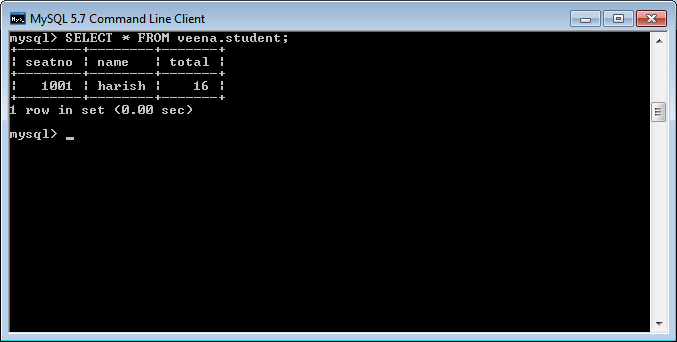












**RESULT:**

Thus a simple student database management system using event-driven and concurrent programming paradigms of Java using JDBC to connect a back-end database has been executed successfully.