Lab	Program

1.	a. A demo on NetBeans IDE.
	b. Write a java Program to print “Hello World”.
	
	public class p1b {
	public static void main(String[] args) {
	System.out.println("Hello world!");
	}
	}
	c. Write a java Program to print Address in multiple lines.
	
	public class p1c {
	public static void main(String[] args) {
	System.out.println("\" Jay \",\nVivekanand-nagar:5,\nDevpara Chowk,\nRajkot 360002.");
	}
	}

2.	a. Write a console program to define and initialize a variable of type byte to 1, and then successively multiply it by 2 and display its value 8 times. Explain the reason for the last result.
	
	public class p2a {
	public static void main(String[] args) {
	byte a=1;
	int i=1;
	while(i<=8)
	{
	a=(byte)(a*2);
	System.out.println(a);
	i++;
	}
	}
	}
	
	b. Write a program that defines a floating-point variable initialized with a dollar value for your income and a second floating-point variable initialized with a value corresponding to a tax rate of 35 percent. Calculate and output the amount of tax you must pay with the Rs. and paisa stored as separate integer values (use two variables of type int to hold the tax, perhaps taxRs and taxPaisa).
	
	public class p2b {
	public static void main(String[] args) {
	int taxrs,taxpaisa,t;float a,b,rs;
	a=(float)300.85;
	b=(float)((25.00*a)/200.00);
	rs=(float)(b*85.30);
	taxrs=(int)rs;
	System.out.println("Tax Rate in Rs="+taxrs);
	taxpaisa=(int) ((rs-taxrs)*100);
	System.out.println("Tax rate in dollar ="+b);
	System.out.println("Tax Rate in paisa ="+taxpaisa);
	}
	}

3.	a. Write a program that calculate percentage marks of the student if marks of 6 subjects are given.
	
	public class p3a {
	public static void main(String[] args) {
	int ps=98,oop=99,wt=97,ds=95,pp=96,dbms=99;

```
        float per;
        System.out.println("SUBJECT MARKS");
        System.out.println("PS = "+ps);
        System.out.println("OOP = "+oop);
        System.out.println("WT = "+wt);
        System.out.println("DS = "+ds);
        System.out.println("PP = "+pp);
        System.out.println("DBMS = "+dbms);
        per=(float) ((pp+ps+oop+wt+ds+dbms)*100)/600;
        System.out.println("PERCENTAGE ="+per);
    }
}
```

b. Write a program to display a random choice from a set of six choices for breakfast (you could use any set; for example, scrambled eggs, waffles, fruit, cereal, toast, or yogurt).

```
import java.util.Random;
public class p3b {
   public static void main(String[] args) {
      Random re=new Random();
      int r;
      r= re.nextInt(6);
      System.out.println("Random number="+r);
      System.out.println("1.Scrambled eggs");
      System.out.println("2.Waffles");
      System.out.println("3.Fruit");
      System.out.println("4.cereal");
      System.out.println("5.toast");
      System.out.println("6.yogurt");
      switch (r)
      {
         case 1:
            System.out.println("your choice is =Scrambled eggs");
            break;
         case 2:
            System.out.println("your choice is =Waffles");
            break;
         case 3:
            System.out.println("your choice is =Fruit");
            break;
         case 4:
            System.out.println("your choice is =cereal");
            break;
         case 5:
            System.out.println("your choice is =toast");
            break;
         case 6:
            System.out.println("your choice is =yogurt");
            break;
         default:
            System.out.println("Invalid choice..");break;
      }
   }
}
```

4. a. When testing whether an integer is a prime, it is sufficient to try to divide by integers up to the square root of the number being tested. Write a program to use this approach.

```
import java.util.*;
public class p4a {
   public static void main(String[] args) {
      int n,sq,f=0,i;
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter no:");
      n=sc.nextInt();
      sq=(int)(Math.sqrt(n));
      for (i=sq;i<n;i++)
      {
         if((n%i)==0){
            f=1;break;
         }
      }
      if(f==1) System.out.println(n+" is not prime");
      else System.out.println(n+" is prime");
   }
}
```

b. A lottery requires that you select six different numbers from the integers 1 to 49. Write a program to do this for you and generate five sets of entries.

```
public class p4b {
   public static void main(String[] args) {
      int lucky,i,f=0;
      int a[]=new int[6];
      System.out.println("Ticket No");
      for(i=0;i<6;i++)
      {
         a[i]=(int)((6*Math.random())+1);
      }
      lucky=(int)((49*Math.random())+1);
      for(i=0;i<6;i++)
      {
         System.out.println(+a[i]);
      }
      System.out.println("Lucky Number: "+lucky);
      for(i=0;i<6;i++)
      {
         if(a[i]==lucky)
         {
            f=1;
            break;
         }
      }
      if(f==1)
      {
         System.out.println(lucky+" no ticket is winner");
      }
      else
      {
         System.out.println("Sorry today no one is winner");
```

```
        }
    }
}
```

5. a. Write a program to generate a random sequence of capital letters that does not include vowels.

```
import java.util.Random;
public class p5a {
   public static void main(String a[])
   {
      int i;
      Random r = new Random();
      char c[] = new char[5];
      for(i=0;i<5;i++)
      {
         c[i]=(char)(r.nextInt(26) + 'A');
      }
      for(i=0;i<5;i++)
      {
         switch(c[i])
         {
            case 'A':
               System.out.println("Sorry this letter is vowel");
               break;
            case 'E':
               System.out.println("Sorry this letter is vowel");
               break;
            case 'I':
               System.out.println("Sorry this letter is vowel");
               break;
            case 'O':
               System.out.println("Sorry this letter is vowel");
               break;
            case 'U':
               System.out.println("Sorry this letter is vowel");
               break;
            default:
               System.out.println(c[i]);
               break;
         }
      }}
}
```

b. Write an interactive program to print a string entered in a pyramid form. For instance, the string "stream" has to be displayed as follows:

```
        S
       S t
      S t r
     S t r e
    S t r e a
   S t r e a m
```

```
import
```

```
java.util.*;
public class
p5b {

      public static void main(String[] args){
      char c;
      int i,j;
      String s="Stream";
      int k;
      for(i=0;i<s.length();i++)
      {
         for(k=0;k<s.length()-i;k++) {
            System.out.print(" ");
         }

         for(j=0;j<=i;j++)
         {

c=s.charAt(j);
            System.out.print(c+" ");
         }
         System.out.println(" ");
      }

   }
}
```

c. Create an array of String variables and initialize the array with the names of the months from January to December. Create an array containing 12 random decimal values between 0.0 and 100.0. Display the names of each month along with the corresponding decimal value. Calculate and display the average of the 12 decimal values.

```
public class p5c {
   public static void main(String[] args) {
      String[]
mon={"January","February","March","April","May","June","July","August","September","October","No
vember","December"};
      int i;
      double sum=0;
      double n[]=new double[12];
      System.out.println("Random number");
      for(i=0;i<12;i++)
      {
         n[i]=((100.00*Math.random())+1);
      }
      for(i=0;i<12;i++)
      {
         System.out.format("%.2f ",n[i]);
         System.out.println(mon[i]);
         sum=sum+n[i];
      }
      System.out.format("Sum of 12 number :%.2f\n",sum);
      System.out.format("Average :%.2f",(sum/12));
   }
}
```

6. a. Write a program to accept a line and check how many consonants and vowels are there in line.

```
import java.util.*;
public class p6a {
   public static void main(String args[]){
      int vCount = 0, cCount = 0;
      Scanner sc =new Scanner(System.in);
      System.out.println("Enter String: ");
      String str =sc.nextLine();
      str = str.toLowerCase();
      for(int i = 0; i < str.length(); i++) {
         if(str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i' || str.charAt(i) == 'o' ||
str.charAt(i) == 'u') {
            vCount++;
         }
         else if(str.charAt(i) >= 'a' && str.charAt(i)<='z') {
            cCount++;
         }
      }
      System.out.println("Number of vowels: " + vCount);
      System.out.println("Number of consonants: " + cCount);
   }
}
```

b. Write a program to find length of string and print second half of the string.

```
import java.util.*;
public class p6b {
   public static void main(String[] args){
      Scanner sc=new Scanner(System.in);
      String s;
      System.out.println("Enter a String :");
      s=sc.nextLine();
      int l=s.length();
      System.out.println("Length of the string is "+l);
      System.out.println("Your sub String is:"+s.substring(l/2));
   }
}
```

c. Write a program to find that given number or string is palindrome or not.

```
import java.util.*;
public class p6c {
   public static void main(String args[])
   {
      String original, reverse = "";
      Scanner in = new Scanner(System.in);
      System.out.println("Enter a string/number to check if it is a palindrome:");
      original = in.nextLine();
      int length = original.length();
      for ( int i = length - 1; i >= 0; i-- )
         reverse = reverse + original.charAt(i);
      if (original.equals(reverse))
         System.out.println("Entered string/number is a palindrome.");
      else
```

```
            System.out.println("Entered string/number isn't a palindrome.");
        }
    }
```

7. a. Create a class which ask the user to enter a sentence, and it should display count of each vowel type in the sentence. The program should continue till user enters a word “quit”. Display the total count of each vowel for all sentences.

```
import java.util.*;
public class p7a {
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        int a=0,e=0,i=0,o=0,u=0;
        String s;
        System.out.println("Enter Lines:");
        while(true)
        {
            s = sc.nextLine();
            int la=0,le=0,li=0,lo=0,lu=0;
            if(s.equals("quit"))break;
            for(int x=0;x<s.length();x++)
            {
                switch(s.charAt(x))
                {
                    case 'A': case 'a':la++;a++;break;
                    case 'E': case 'e':le++;e++;break;
                    case 'I': case 'i':li++;i++;break;
                    case 'O': case 'o':lo++;o++;break;
                    case 'U': case 'u':lu++;u++;break;
                }
            }
            System.out.println("Vowel in sentence is:");
            System.out.println("A:"+la);
            System.out.println("E:"+le);
            System.out.println("I:"+li);
            System.out.println("O:"+lo);
            System.out.println("U:"+lu);
        }
        System.out.println("Total Vowel is:");
        System.out.println("A:"+a);
        System.out.println("E:"+e);
        System.out.println("I:"+i);
        System.out.println("O:"+o);
        System.out.println("U:"+u);
    }
}
```

b. Write a program to create a class Student with data ‘name, city and age’ along with method printData to display the data. Create the two objects s1, s2 to declare and access the values.

```
import java.util.*;
class Student {
    String name;
    String city;
```

```
    int age;
    void printfData(){
      System.out.println("NAME:"+name);
      System.out.println("CITY:"+city);
      System.out.println("AGE:"+age);
    }
  }
  public class p7b {
    public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      Student s1 = new Student();
      Student s2 = new Student();
      s1.name=sc.nextLine();
      s2.name=sc.nextLine();
      s1.city=sc.nextLine();
      s2.city=sc.nextLine();
      s1.age=sc.nextInt();
      s2.age=sc.nextInt();
      s1.printfData();
      s2.printfData();
    }
  }
```

c. Write a program to create a class Student2 along with two method getData(), printData() to get the value through argument and display the data in printData. Create the two objects s1, s2 to declare and access the values.

```
import java.util.*;
class Student1 {
  String name;
  String city;
  int age;
  void getData(String name,String city,int age){
    this.name=name;
    this.city=city;
    this.age=age;
  }
  void printfData(){
    System.out.println("NAME:"+name);
    System.out.println("CITY:"+city);
    System.out.println("AGE:"+age);
  }
}
public class p7c {
  public static void main(String[] args) {
    Student1 s1 = new Student1();
    Student1 s2 = new Student1();
    s1.getData("JAY","BHATT",20);
    s2.getData("MSD","RANCHI",41);
```

```
        s1.printfData();
        s2.printfData();
    }
}
```

d. WAP using parameterized constructor with two parameters id and name. While creating the objects obj1 and obj2. Pass two arguments so that this constructor gets invoked after creation of obj1 and obj2. Create one method to print the data.

```
import java.util.*;
class Student2 {
    int ID;
    String name;
    Student2(String name,int ID){
        this.ID=ID;
        this.name=name;
    }
    void printfData(){
        System.out.println("NAME:"+name);
        System.out.println("ID:"+ID);
    }
}
public class p7d {
    public static void main(String[] args) {
        Student2 s1 = new Student2("JAY",513);
        Student2 s2 = new Student2("MSD",781);
        s1.printfData();
        s2.printfData();
    }
}
```

e. Write a program in JAVA to create a class Bird also declares the different parameterized constructor to display the name of Birds.

```
class Bird{
public Bird(String  a) {
        System.out.println(a);
        }

public Bird(String  a,String  b,String  c) {
    System.out.println(a);
    System.out.println(b);
    System.out.println(c);
        }
    public Bird() {
        System.out.println("No parameters.");
    }
public Bird(String  a,String  b) {
    System.out.println(a);
    System.out.println(b);
```

```
        }
        }
public class p7e {
    public static void main(String[] args) {
        Bird b1 = new Bird();
        Bird b2 = new Bird("Peacock");
        Bird b3 = new Bird("Parrot","Owl");
        Bird b4 = new Bird("Bald eagle","Crow","Bat");
    }

}
```

8. a. Write a program in java to generate an abstract class A also class B inherits the class A. generate the object for class B and display the text “call me from B”.

```
abstract class A{
    abstract void display();
}
class B extends A{
    @Override
    void display() {
        System.out.println("Call me from B");
    }
}
public class p8a {
    public static void main(String[] args) {
        B obj = new B();
        obj.display();
    }
}
```

b. Write a java program in which you will declare two interface sum and Add inherits these interface through class A1 and display their content.

```
interface Sum{
    public void displaySum();
}
interface Add{
    public void displayAdd();
}
class A1 implements Sum,Add{
    public void displaySum(){
        System.out.println("This is Sum interface");
    }
    public void displayAdd(){
        System.out.println("This is Add interface");
    }
}
public class p8b {
    public static void main(String[] args) {
        A1 obj = new A1();
```

```
        obj.displaySum();
        obj.displayAdd();
    }
}
```

9. a. Write a java program in which you will declare an abstract class Vehicle inherits this class from two classes car and truck using the method engine in both display “car has good engine” and “truck has bad engine”.

```
abstract class Vehicle{
   abstract void display();
}
class Car extends Vehicle{
   void display(){
      System.out.println("car has good engine");
   }
}
class Truck extends Vehicle{
   void display(){
      System.out.println("Truck has bad engine");
   }
}
public class p9a {
   public static void main(String[] args) {
      Car c = new Car();
      Truck t = new Truck();
      c.display();
      t.display();
   }
}
```

b. Define an abstract base class Shape that includes protected data members for the (x, y) position of a shape, a public method to move a shape, and a public abstract method show() to output a shape. Derive subclasses for lines, circles, and rectangles. You can represent a line as two points, a circle as a center and a radius, and a rectangle as two points on diagonally opposite corners. Test the classes by objects of the derived classes, and then invoking the show() method.

```
abstract class Shape{
   protected double x1 = 2, y1 = 5;
   protected double x2 = 6, y2 = 10;
   abstract void Show();
}
class Lines extends Shape{
   @Override
   void Show() {
      System.out.println("Points of Line is :("+x1+","+y1+") ("+x2+","+y2+")");
   }
}
class Circles extends Shape{
   @Override
   void Show() {
      System.out.println("Center of Circle is "+x1+" Radius of Circle is "+y1);
   }
}
class Rectangle extends Shape{
   @Override
   void Show() {
      System.out.println("Points of Diagonal of Rectangle is :("+x1+","+y1+") ("+x2+","+y2+")");
```

```
    }
}
public class Main {
    public static void main(String[] args) {
        Lines L = new Lines();
        Circles C = new Circles();
        Rectangle R = new Rectangle();
        L.Show();
        C.Show();
        R.Show();
    }
}
```

10. a. Write a program that will generate exceptions of type NullPointerException, NegativeArraySizeException, and IndexOutOfBoundsException. Record the catching of each exception by displaying the message stored in the exception object.

```
class Exceptions{
    void NullPointerException(){
        String s = null;
        try{
            if(s.length()>0)
                System.out.println("Yes");
            else
                System.out.println("No");
        }
        catch (NullPointerException e){
            System.out.println(e);
        }
    }
    void NegativeArraySizeException(){
        try{
            int [] n = new int [-1];
        }
        catch (NegativeArraySizeException e){
            System.out.println(e);
        }
    }
    void IndexOutOfBoundsException(){
        int [] n = new int [5];
        try{
            System.out.println(n[5]);
        }
        catch (IndexOutOfBoundsException e){
            System.out.println(e);
        }
    }
}
public class p10a {
    public static void main(String[] args) {
        Exceptions e = new Exceptions();
        e.NullPointerException();
        e.NegativeArraySizeException();
        e.IndexOutOfBoundsException();
```

```
    }
  }
```

b. Write a program that calls a method that throws an exception of type ArithmeticException at a random iteration in for loop. Catch the exception in the method and pass the iteration count when the exception occurred to the calling method by using an object of an exception class you define.

```
class arExp{
  void throwException(){
    int a = 5;
    try{
      for(int i=5;i>=0;i--){
        System.out.println(a/i);
      }
    }
    catch (Exception e){
      System.out.println(e);
    }
  }
}
public class Main {
  public static void main(String[] args) {
    arExp a = new arExp();
    a.throwException();
  }
}
```

11. a. Create three threads in synchronization that will reduce the input number by one every time and displays the output as follows:

Start thread_1: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 End
Start thread_2: 10 9 8 7 6 5 4 3 2 1 End
Start thread_3: 5 4 3 2 1 End

```
class MyThread1 extends Thread{

synchronized public void run() {
  for(int i=15; i>=1; i--){
    System.out.println(i);
  }
System.out.println("END");
      }     }

class MyThread2 extends Thread{

synchronized public void run() {
  for(int i=10; i>=1; i--){
    System.out.println(i);
  }
System.out.println("END");
      }     }

class MyThread3 extends Thread{
```

```
        synchronized public void run() {
          for(int i=5; i>=1; i--){
            System.out.println(i);
          }
        System.out.println("END");
              }       }
          class Main{
          public static void main(String args[]){

          MyThread1 t1=new MyThread1();
          MyThread2 t2=new MyThread2();
          MyThread3 t3=new MyThread3();
        t1.start();
        t2.start();
        t3.start();
          }   }
```

b. Write a program to demonstrate Inter thread Communication.

```
class Customer
{
int amount=100;
 synchronized void withdraw(int amount)
{
System.out.println("Going to withdraw...");

if(this.amount<amount){
System.out.println("Less balance; waiting for deposit...");
try{wait();}catch(Exception e){}
   }
this.amount-=amount;
System.out.println("Withdraw of " +amount+" completed...");
}

synchronized void deposit(int amount)
{
System.out.println("going to deposit...");
this.amount+=amount;
System.out.println("Deposit of " + amount+ " completed... ");
notify();
}
}

class InterThread
{
public static void main(String args[]){
final Customer c=new Customer();
new Thread(){
public void run(){c.withdraw(500);}
   }.start();

new Thread(){
public void run(){c.deposit(12000);}
   }.start();          } }
```

c. Write a program to demonstrate Deadlock.

```
public class Main {
  public static Object Lock1 = new Object();
  public static Object Lock2 = new Object();

  public static void main(String args[]) {
    ThreadDemo1 T1 = new ThreadDemo1();
    ThreadDemo2 T2 = new ThreadDemo2();
    T1.start();
    T2.start();
  }

  private static class ThreadDemo1 extends Thread {
    public void run() {
      synchronized (Lock1) {
        System.out.println("Thread 1: Holding lock 1...");

        try { Thread.sleep(10); }
        catch (InterruptedException e) {}
        System.out.println("Thread 1: Waiting for lock 2...");

        synchronized (Lock2) {
          System.out.println("Thread 1: Holding lock 1 & 2...");
        }
      }
    }
  }
  private static class ThreadDemo2 extends Thread {
    public void run() {
      synchronized (Lock2) {
        System.out.println("Thread 2: Holding lock 2...");

        try { Thread.sleep(10); }
        catch (InterruptedException e) {}
        System.out.println("Thread 2: Waiting for lock 1...");

        synchronized (Lock1) {
          System.out.println("Thread 2: Holding lock 1 & 2...");
        }
      }
    }
  }
}
```

12. a. WAP Write a main() method that takes the name of a text file as a command line argument and prints every line in lower case.

```
import java.io.*;
class Main
{
 public static void main(String s[])throws Exception
 {
  FileInputStream in =new FileInputStream("x.txt");
  int size=in.available();
  int i;
  char c;
```

```
 for(i=0;i<size;i++)
 {
  c=(char)in.read();
  System.out.print(Character.toLowerCase(c));
 }
 in.close();
}
}
```

b. Write a main() method that counts the number of words in a text file whose name is accepted from standard input. Also print the size of a file.

```
import java.util.*;
import java.io.*;
class Filecount
{
 public static void main(String s[])throws Exception
 {
  FileInputStream in =new FileInputStream("x.txt");
  Scanner sc=new Scanner(in);
  int size=in.available();
  int count=0;
  while(sc.hasNext())
  {
   sc.next();
   count++;
  }
  System.out.println("Size of File: "+size);
  System.out.println("There are "+count+" Word in file");
  in.close();
 }
}
```

c. Write a program using BufferedInputStream, FileInputStream, BufferedOutputStream, and FileOutputStream to copy Content of one file File1.txt into another file File2.txt.

```
import java.io.*;
class Filecopy
{
 public static void main(String s[])throws Exception
 {
  FileInputStream in = new FileInputStream("File1.txt");
  BufferedInputStream bin = new BufferedInputStream(in);
  FileOutputStream ou = new FileOutputStream("File2.txt");
  BufferedOutputStream bou = new BufferedOutputStream(ou);
  int size=in.available();
  int i;
  for(i=0;i<size;i++)
  {
   bou.write((char)bin.read());
  }
  System.out.println("Content of one file is copied to another fle");
  bou.close();
```

```
 ou.close();
 bin.close();
 in.close();
}

}
```

13. a. Write a program to demonstrate ArrayList class. Perform insert, delete and access operation.

```
import java.util.*;
public class Main{
public static void main(String args[]){
ArrayList<String> al=new ArrayList<String>(); //ArrayList al = new ArrayList();
al.add("Ravi");
al.add("Vijay");
al.add("Aarav");
al.add("Ajay");
for(String s2:al)
System.out.println("Elements:"+s2);

al.remove("Ravi");
for(String s:al)
System.out.println("After remove:"+s);

System.out.println("Access index 1:"+ al.get(1));

al.clear();
System.out.println(al.size());
} }
```

b. Write a Program to demonstrate Vector class. Perform insert, delete and access operation.

```
import java.util.*;
public class Main{
public static void main(String args[]){
Vector<String> al=new Vector<String>();
al.add("Ravi");
al.add("Vijay");
al.add("Aarav");
al.add("Ajay");
for(String s2:al)
System.out.println("Elements:"+s2);

al.remove("Ravi");
for(String s:al)
System.out.println("After remove:"+s);

System.out.println("Access index 1:"+ al.get(1));

al.clear();
System.out.println(al.size());
} }
```

14. a. WAP to create and import a Package.

STEP 1: Create a folder p1 and follow the below code,

```
package p1;
public class testpackage
{
public void display()
{
System.out.println("Hi");
}
}
```

Save the file in p1 folder as “testpackage” and compile it.

STEP 2: Come out from the folder p1 and save the below code as “test”.

```
import p2.*;
public class test
{
public static void main(String args[])
{
    testpackage tp = new testpackage();
    tp.display();
}
}
```

Now compile and run the “test” file

b. WAP to demonstrate List, Stack and Maps classes and its operations.

LinkedList:

```
import java.util.*;
public class Main{
public static void main(String args[]){
LinkedList<String> al=new LinkedList<String>();
al.add("Ravi");
al.add("Vijay");
System.out.println(al);
al.addFirst("Zumba");
System.out.println("After addFirst:"+al);
al.removeFirst();
System.out.println("After removeFirst:"+al);
} }
```

Stack:

```
import java.util.*;

class Main {
public static void main(String[] args)
{
    Stack stack1 = new Stack();
    stack1.push(4);
    stack1.push("All");
    stack1.push("Hello");

    System.out.println(stack1);

    stack1.pop();
```

```
        System.out.println(stack1);

        System.out.println(stack1.peek());
    } }

    Map:
    import java.util.*;
    public class Main {
    public static void main(String[] args) {

    Map<String, String> map = new LinkedHashMap<>();

    map.put("123", "John Smith");
    map.put("111", "George Smith");
    map.put("123", "Steve Yao");
    map.put("222", "Steve Yao");
    System.out.println(map);

    System.out.println(map.get("222"));

    map.clear();

    System.out.println("Size:"+map.size());

    }
    }
```

c. A mini project demo on a) Brick Breaker Game b) Notepad application.