**Experiment No: 4.1**

**Aim:**

|  |
| --- |
| Develop a Name database. (use Vector to store the database) User Menu should include: a) Add Names. b) Search Names. c) Delete Names. d) View All Names. e) Exit from the application. |

**Code:**

|  |
| --- |
| **import** java.util.Scanner;  **import** java.util.Vector;  **class** b {  Scanner ip=**new** Scanner(System.***in***);  String st;  Vector v=**new** Vector();  **void** vector(){  **while**(**true**){  System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nMenu\na)ADD NAMES\nb)DELETE NAMES\nc)SEARCH NAMES\nd)SHOW NAMES\ne)EXIT\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.***out***.println("Enter your choice:");  **char** ch=ip.next().charAt(0);  **switch**(ch){  **case** 'a':  add\_name();  **break**;  **case** 'b':  del\_name();  **break**;  **case** 'c':  search();  **break**;  **case** 'd':  show\_name();  **break**;  **case** 'e':  **return**;  **default**:  System.***out***.println("Wrong choice");  } } }  **void** add\_name(){  System.***out***.println("Enter name");  st=ip.next();  v.add(st);  }  **void** show\_name(){  **int** size=v.size();  System.***out***.println("Enetered names are");  **for**(**int** i=0;i<size;i++){  System.***out***.println(v.elementAt(i));  }  }  **void** del\_name(){  System.***out***.println("Enter then name to be deleted");  String aa =ip.next();  v.remove(aa);  }  **void** search(){  System.***out***.println("Enter the name to be Searched");  String a =ip.next();  **boolean** z=v.contains(a);  **if** (z==**true**)  System.***out***.println("Entered name is present");  **else**    System.***out***.println("Entered name not found");  }  }  **public** **class** A{  **public** **static** **void** main(String args[]) {  b object=**new** b();  object.vector();  }  } |

**Output:**

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Menu  a)ADD NAMES  b)DELETE NAMES  c)SEARCH NAMES  d)SHOW NAMES  e)EXIT  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Enter your choice:  a  Enter name  HIMANSHU  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Menu  a)ADD NAMES  b)DELETE NAMES  c)SEARCH NAMES  d)SHOW NAMES  e)EXIT  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Enter your choice:  a  Enter name  LAHARE  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Menu  a)ADD NAMES  b)DELETE NAMES  c)SEARCH NAMES  d)SHOW NAMES  e)EXIT  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Enter your choice:  b  Enter then name to be deleted  LAHARE  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Menu  a)ADD NAMES  b)DELETE NAMES  c)SEARCH NAMES  d)SHOW NAMES  e)EXIT  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Enter your choice:  d  Enetered names are  HIMANSHU  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Menu  a)ADD NAMES  b)DELETE NAMES  c)SEARCH NAMES  d)SHOW NAMES  e)EXIT  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Enter your choice:  e |

**Experiment No: 4.2**

**Aim:**

|  |
| --- |
| Create a package P1 and inside which add two classes 'Fibo' and 'Fact'. Complete the following program: import P1.Fibo; import P1.Fact; public class NewClass { public static void main(String arg[]) { Fibo f1=new Fibo(5); f1.show();//should show fibonacci series upto 5 terms. Fact f2=new Fact(5); f2.display();//should display the factorial of 5 } } |

**Code:**

|  |
| --- |
| **package** p1;  **public** **class** fact {  **static** **int** *n*;  **public** fact(**int** i){  *n*=i;  }  **public** **static** **void** display() {  **int** fact=1;  **while**(*n*>0) {  fact=*n*\*fact;  *n*--;    }  System.***out***.println("FACTORIAL IS "+fact);  }  } System.***out***.println("method in g");  }  f obj1=**new** f();  }  }  **class** h{  **public** **static** **void** main(String as[]){  f.g object1=**new** f().**new** g();  object1.methoding();  }  }  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  **package** p1;  **public** **class** fibo {  **static** **int** *n*;  **public** fibo(**int** i){  *n*=i;  }  **public** **static** **void** show() {  **int** i=0;  **int** a=0;  **int** b=1;  **int** c;  System.***out***.print("0\t1\t");  **while**(i<*n*) {  c=a+b;  System.***out***.print(c+"\t");  a=b;  b=c;  i++;  }  }  }  package p2;  import p1.fact;  import p1.fibo;  public class newCLas {  public static void main(String arg[])  {  p1.fibo op=new p1.fibo(5);  op.show();  p1.fact op1=new p1.fact(5);  op1.display();  }  } |

**Output:**

|  |
| --- |
| 0 1 1 2 3 5 8 FACTORIAL IS 120 |

**Experiment No: 4.3**

**Aim:**

|  |
| --- |
| Create a Shape Interface which has a member method area(). Derive two subclasses Circle and Triangle from it. Using reference of Shape class fill the required members in Circle and Triangle also display the area of Circle and Triangle. Take input from user while filling data members. |

**Code:**

|  |
| --- |
| **import** java.util.\*;  **interface** shape{  **void** area();  **static** **double** ***pi***=3.14;  }  **class** circle **implements** shape{  **public** **void** area(){  Scanner ip = **new** Scanner(System.***in***);  System.***out***.println("Enter the radius:");  **double** ra=ip.nextInt();  **double** area=***pi***\*ra\*ra;  System.***out***.println("Area of circle"+area);  }  }  **class** triangle **implements** shape{  **public** **void** area(){  Scanner ip = **new** Scanner(System.***in***);  System.***out***.println("Enter base:");  **double** h=ip.nextInt();  System.***out***.println("Enter height:");  **double** b=ip.nextInt();  **double** area=0.5\*h\*b;  System.***out***.println("Area is triangle"+area);  }  }  **public** **class** test {  **public** **static** **void** main(String args[]){  circle ob=**new** circle();  ob.area();  triangle ob1=**new** triangle();  ob1.area();    }} |

**Output:**

|  |
| --- |
| Enter the radius:  7  Area of circle153.86  Enter base:  5  Enter height:  6  Area is triangle15.0 |

**Experiment No: 4.4**

**Aim:**

|  |
| --- |
| Write a program to demonstrate the effect of access modifiers (default, protected, public and private) on members with and without inheritance within a package and outside a package. |

**Code:**

|  |
| --- |
| **package** pak;  **public** **class** A {  **public** **static** **void** m1() {  System.***out***.println("class a.public ");  }  **private** **static** **void** m2() {  System.***out***.println("class a.private");  }  **protected** **static** **void** m3() {  System.***out***.println("class a.protected");  }  **static** **void** m4() {  System.***out***.println("class a.default");  }    }  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  **package** pak;  **public** **class** B {  **public** **static** **void** main(String[] as) {  A ob= **new** A();  ob.*m1*();  ob.*m2*(); //PRIVATE not visible  ob.*m3*();  ob.*m4*();  }}  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  **package** pak;  **public** **class** C **extends** A{  **public** **static** **void** main() {  A ob=**new** A();  ob.*m1*();  ob.*m2*(); //PRIVATE not visible  ob.*m3*();  ob.*m4*();  }  }  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  **package** pak1;  **import** pak.A;  **public** **class** D {  **public** **static** **void** main(String [] as) {  A ob1=**new** A();  ob1.*m1*();// VISIBLE PUBLIC METHODS  ob1.*m2*();//NOT VISIBLE PRIVATE METHOD  ob1.*m3*();//NOT VISIBLE PROTECTED METHOD  ob1.*m4*();//NOT VISIBLE DEFUALT METHOD  }  }  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  **package** pak1;  **import** pak.\*;;  **public** **class** E **extends** A{  **public** **static** **void** main(String[] as) {  A ob1=**new** A();  ob1.*m1*(); *m1*();// VISIBLE PUBLIC METHODS  ob1.*m2*(); //NOT VISIBLE PRIVATE METHOD  ob1.*m3*();*m3*();//VISIBLE PROTECTED ON INHERITANCE  ob1.*m4*();//NOT VISIBLE DEFUALT METHOD  }  } |

**Output:**

|  |
| --- |
| Non visible methods will cause compilation error |

**Experiment No: 4.5**

**Aim:**

|  |
| --- |
| Write a program to show inbuilt checked and unchecked exception |

**Code:**

|  |
| --- |
| **class** A{ //checked execption error shows during compilation  **static** **int** *a*=100;  **static** **int** *b*=0;  **static** **int** *q*;  **static** **void** dvde() **throws** ArithmeticException{//checked exception  *q*=*a*/*b*;  }  **public** **static** **void** main(String as[]) {  **try** {  *dvde*();  }  **catch**(ArithmeticException w) {  System.***out***.println("divison is "+w);  B object=**new** B();  object.uncheck();  }}  }  **class** B { //unchecked exception no compilation error(runtime errror)  **void** uncheck() **throws** ArrayIndexOutOfBoundsException  {  **int** arr[] ={1,2,3,4,5};  System.***out***.println(arr[7]);  }  } |

**Output:**

|  |
| --- |
| divison is java.lang.ArithmeticException: / by zero  Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 7 out of bounds for length 5  at B.uncheck(A.java:23)  at A.main(A.java:15) |

**Experiment No: 4.6**

**Aim:**

|  |
| --- |
| Write a program to demonstrate the two ways in which singleton classes can be created and use factory methods from those classes to create instance of the singleton classes |

**Code:**

|  |
| --- |
| **public** **class** A {  **public** **static** **void** main(String as[]) {  Singleton ob=Singleton.*getInstance*();  ob.*setn*(121);  System.***out***.println(Singleton.*getn*());  Singleton1 ob1=Singleton1.*getInstance*();  ob1.*setn*(1221);  System.***out***.println(Singleton1.*getn*());  }}  **class** Singleton {  **static** **int** *n*;  **static** **int** getn() {  **return** *n*;  }  **static** **void** setn(**int** a) {  *n*=a;  }  **private** **static** Singleton *ob* = **null**;  **private** Singleton() {}  **public** **static** Singleton getInstance() {  **if** ( *ob*== **null**) {  *ob*= **new** Singleton();  }  **return** *ob*;  }  }  **class** Singleton1 {  **static** **int** *n*;  **static** **int** getn() {  **return** *n*;  }  **static** **void** setn(**int** a) {  *n*=a;  }  **private** **static** **final** Singleton1 ***o*** = **new** Singleton1();  **private** Singleton1() {}  **public** **static** Singleton1 getInstance() {  **return** ***o***;  } } |

**Output:**

|  |
| --- |
| 121  1221 |

**Date : 16/08/2019**

**Experiment No: 4.7**

**Aim:**

|  |
| --- |
| Class A is a singleton class. Complete the following program: public class NewClass { public static void main(String arg[]) { A ob1,ob2,ob3; // objects can be declared but new instances cannot be created ob1=A.getA(); //getA() is a factory method ob2=A.getA(); ob3=A.getA(); ob1.setN(100); System.out.println(ob1.getN()+" "+ob2.getN()+" "+ob3.getN()); ob2.setN(200); System.out.println(ob1.getN()+" "+ob2.getN()+" "+ob3.getN()); ob3.setN(300); System.out.println(ob1.getN()+" "+ob2.getN()+" "+ob3.getN()); } } Output: 100 100 100 200 200 200 300 300 300 |

**Code:**

|  |
| --- |
| **class** A{  **static** **int** *n*;  **private** A() {}  **public** **static** A getA() {  A ob1=**new** A();  **return** ob1;  }  **public** **static** **void** setN(**int** a) {  *n*=a;  }  **public** **static** **int** getN() {  **return** *n*;  }  }  **public** **class** NewClass {  **public** **static** **void** main(String arg[]) {  A ob1,ob2,ob3; // objects can be declared but new instances cannot be created  ob1=A.*getA*(); //getA() is a factory method  ob2=A.*getA*();  ob3=A.*getA*();  ob1.*setN*(100);  System.***out***.println(ob1.*getN*()+" "+ob2.*getN*()+" "+ob3.*getN*());  ob2.*setN*(200);  System.***out***.println(ob1.*getN*()+" "+ob2.*getN*()+" "+ob3.*getN*());  ob3.*setN*(300);  System.***out***.println(ob1.*getN*()+" "+ob2.*getN*()+" "+ob3.*getN*());  }  } |

**Output:**

|  |
| --- |
| 100 100 100  200 200 200  300 300 300 |

**Experiment No: 4.8**

**Aim:**

|  |
| --- |
| Write a program to generate a customized un-checked exception whenever a number input from the user is negative |

**Code:**

|  |
| --- |
| **import** java.util.Scanner;  **class** dd **extends** RuntimeException{  dd(String ss){  **super**(ss);  }  }  **public** **class** A {  **static** **int** *q*;  **public** **static** **void** main(String[] as) {  Scanner op=**new** Scanner(System.***in***);  System.***out***.println("ENter a number");  *q*=op.nextInt();  **try**{    **if** (*q*<0 ) {  **throw** **new** dd("INVALID NUMBER");  }    **else**  System.***out***.println("NUMBER IS VALID");  }  **catch**(dd e) {  System.***out***.println(e.getMessage());  }  }  } |

**Output:**

|  |
| --- |
| A is30  B is20  C is10 |

**Experiment No: 4.9**

**Aim:**

|  |
| --- |
| Write a program to generate a customized checked exception whenever a number input from the user is either less than 20 or greater than 50. |

**Code:**

|  |
| --- |
| **import** java.util.Scanner;  **class** dd **extends** Exception{  dd(String ss){  **super**(ss);  }  }  **public** **class** A {  **static** **int** *q*=30;  **public** **static** **void** main(String[] as) {  Scanner op=**new** Scanner(System.***in***);  System.***out***.println("ENter a numbe");  *q*=op.nextInt();  **try**{  **if** (*q*<20 || *q*>50 ) {  **throw** **new** dd("INVALID NUMBER");  }  **else**  System.***out***.println("NUMBER IS VALID");  }  **catch**(dd e) {  System.***out***.println(e.getMessage());  }  }  } |

**Output:**

|  |
| --- |
| ENter a numbe  59  INVALID NUMBER |