**Task 19**  
Wap to display the content of the above enum.. (main  needs to be added)

Program:-  
  
public enum Element {

H("Hydrogen", 1, 1.008f),

HE("Helium", 2, 4.0026f),

// ...

NE("Neon", 10, 20.180f);

private static final Map<String, Element> BY\_LABEL = new HashMap<>();

private static final Map<Integer, Element> BY\_ATOMIC\_NUMBER = new HashMap<>();

private static final Map<Float, Element> BY\_ATOMIC\_WEIGHT = new HashMap<>();

static {

for (Element e : values()) { //for each loop

BY\_LABEL.put(e.label, e);

BY\_ATOMIC\_NUMBER.put(e.atomicNumber, e);

BY\_ATOMIC\_WEIGHT.put(e.atomicWeight, e);

}

}

public final String label;

public final int atomicNumber;

public final float atomicWeight;

private Element(String label, int atomicNumber, float atomicWeight) {

this.label = label;

this.atomicNumber = atomicNumber;

this.atomicWeight = atomicWeight;

}

public static Element valueOfLabel(String label) {

return BY\_LABEL.get(label);

}

public static Element valueOfAtomicNumber(int number) {

return BY\_ATOMIC\_NUMBER.get(number);

}

public static Element valueOfAtomicWeight(float weight) {

return BY\_ATOMIC\_WEIGHT.get(weight);

}

}

public class ElementTest {

public static void main(String[] args) {

Element e1 = Element.valueOfLabel("Helium");

System.out.println("Element by Label: " + e1); // Should print: HE

Element e2 = Element.valueOfAtomicNumber(1);

System.out.println("Element by Atomic Number: " + e2); // Should print: H

Element e3 = Element.valueOfAtomicWeight(20.180f);

System.out.println("Element by Atomic Weight: " + e3); // Should print: NE

}}

output:-  
Element by Label: HE

Element by Atomic Number: H

Element by Atomic Weight: NE   
  
  
  
**Task 20:**  
  
**package** samplePackage8;

**public** **class** Task20 {

**public** **static** **void** main(String[] args) {

**char**[] Name = {'R', 'a', 'n', 'g', 'a', 'N', 'a', 'r', 'a', 'y', 'a', 'n', 'a'};

System.***out***.print("My name is: ");

System.***out***.println(Name);

**int** n = Name.length;

System.***out***.println("There are " + n + " letters in my name");

System.***out***.println("Each letter in my name:");

**for**(**int** i = 0; i < n; i++) {

System.***out***.println(Name[i]);

}

}}   
  
output:-  
My full name is: Ranga Narayana Srinivas Rao

There are 27 letters (including spaces) in my full name

Each letter in my full name:

R

a

n

g

a

N

a

r

a

y

a

n

a

S

r

i

n

i

v

a

s

R

a

o

**Task 21**public class Task21 {

public static void main(String[] args) {

// initializing array

int[] arr = { 1, 2, 3, 4, 5 };

// size of array

int n = arr.length;

// traversing array

for (int i = 0; i < n; i++)

System.out.print(arr[i] + " ");

}

}  
  
output:-  
1 2 3 4 5  
  
  
 **Task 22**  
class Task22 {

public static void main(String[] args)

{

// declares an Array of integers.

int[] arr;

// allocating memory for 5 integers.

arr = new int[5];

// initialize the elements of the array

// first to last (fifth) element

arr[0] = 10;

arr[1] = 20;

arr[2] = 30;

arr[3] = 40;

arr[4] = 50;

// accessing the elements of the specified array

for (int i = 0; i < arr.length; i++)

System.out.println("Element at index "

+ i + " : " + arr[i]);

}

}  
  
  
  
  
  
output:-

Element at index 0 : 10

Element at index 1 : 20

Element at index 2 : 30

Element at index 3 : 40

Element at index 4 : 50

**Task 23**  
class Student {

public int roll\_no;

public String name;

Student(int roll\_no, String name){

this.roll\_no = roll\_no;

this.name = name;

}

}

public class Task23 {

public static void main(String[] args){

// declares an Array of Student

Student[] arr;

// allocating memory for 5 objects of type Student.

arr = new Student[5];

// initialize the elements of the array

arr[0] = new Student(1, "aman");

arr[1] = new Student(2, "vaibhav");

arr[2] = new Student(3, "shikar");

arr[3] = new Student(4, "dharmesh");

arr[4] = new Student(5, "mohit");

// accessing the elements of the specified array

for (int i = 0; i < arr.length; i++)

System.out.println("Element at " + i + " : { "

+ arr[i].roll\_no + " "

+ arr[i].name + " }");

}

}  
  
output:-  
Element at 0 : { 1 aman }

Element at 1 : { 2 vaibhav }

Element at 2 : { 3 shikar }

Element at 3 : { 4 dharmesh }

Element at 4 : { 5 mohit }  
  
  
**Task 24**class Student {

public String name;

Student(String name) {

this.name = name;

}

@Override

public String toString() {

return name;

}

}

public class Task24 {

public static void main(String[] args) {

// declares an Array and initializing the elements of the array

Student[] myStudents = new Student[]{

new Student("Dharma"), new Student("sanvi"),

new Student("Rupa"), new Student("Ajay")

};

// accessing the elements of the specified array

for (Student m : myStudents) {

System.out.println(m);

}

}

}  
  
  
  
  
  
  
output:-  
Dharma

sanvi

Rupa

Ajay   
  
  
**Task 25**  
public class GFG {

   public static void main(String[] args)

   {

       int[] arr = new int[4];

       arr[0] = 10;

       arr[1] = 20;

       arr[2] = 30;

       arr[3] = 40;

​

       System.out.println(

           "Trying to access element outside the size of array");

       System.out.println(arr[5]);

   }

}

Output:- Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5   
  
  
**Task 26**

import java.io.\*;

​

class GFG {

   public static void main(String[] args){

       // Two Dimensional Array

      // Declared and Initialized

      int[][] arr = new int[3][3];

​

       // Number of Rows

       System.out.println("Rows : " + arr.length);

       // Number of Columns

       System.out.println("Columns : " + arr[0].length);

   }

}

Ouput:- Rows : 3

Columns : 3  
  
  
**Task 27**public class multiDimensional {

     // main function

   public static void main(String args[])

   {

       // declaring and initializing 2D array

       int arr[][] = { { 2, 7, 9 }, { 3, 6, 1 }, { 7, 4, 2 } };

​

       // printing 2D array

       for (int i = 0; i < 3; i++) {

           for (int j = 0; j < 3; j++)

               System.out.print(arr[i][j] + " ");

​

           System.out.println();

       }

   }

}

Output:-  
2 7 9

3 6 1

7 4 2  
  
  
**Task 28**public class Test {

   // Driver method

   public static void main(String args[])

   {

       int arr[] = { 3, 1, 2, 5, 4 };

​

       // passing array to method m1

       sum(arr);

   }

​

   public static void sum(int[] arr)

   {

       // getting sum of array values

       int sum = 0;

​

       for (int i = 0; i < arr.length; i++)

           sum += arr[i];

​

       System.out.println("sum of array values : " + sum);

   }

}

Output:- sum of array values : 15   
  
  
**Task 29**class Test {

   // Driver method

   public static void main(String args[])

   {

       int arr[] = m1();

​

       for (int i = 0; i < arr.length; i++)

           System.out.print(arr[i] + " ");

   }

​

   public static int[] m1()

   {

       // returning  array

       return new int[] { 1, 2, 3 };

   }

}

Output:-1 2 3   
  
  
**Task 30**

**package** samplePackage8;

**public** **class** Test {

**public** **static** **void** main(String args[]) {

**int** intArray[] = { 1, 2, 3 };

**int** cloneArray[] = intArray.clone();

intArray[2] = 4;

System.***out***.println(intArray == cloneArray);

System.***out***.print("intArray: ");

**for** (**int** i = 0; i < intArray.length; i++) {

System.***out***.print(intArray[i] + " ");

}

System.***out***.println();

System.***out***.print("cloneArray: ");

**for** (**int** i = 0; i < cloneArray.length; i++) {

System.***out***.print(cloneArray[i] + " ");

}

}

}

Output:-  
  
false

intArray: 1 2 4

cloneArray: 1 2 3  
  
  
**Task 31**class Test {

public static void main(String args[]) {

int intArray[][] = { { 1, 2, 3 }, { 4, 5 } };

int cloneArray[][] = intArray.clone();

// will print false

System.out.println(intArray == cloneArray);

// will print true as shallow copy is created

// i.e. sub-arrays are shared

System.out.println(intArray[0] == cloneArray[0]);

System.out.println(intArray[1] == cloneArray[1]);

}

}  
  
output:-  
false

true

true

**Task 32  
  
package** samplePackage8;

// No 'public' here, so Task31 is package-private

**class** Task31 {

**int** z;

**public** **void** addition(**int** x, **int** y) {

z = x + y;

System.***out***.println("The sum of the given numbers: " + z);

}

**public** **void** Subtraction(**int** x, **int** y) {

z = x - y;

System.***out***.println("The difference between the given numbers: " + z);

}

}

// This is the public class and the file name must be My\_Calculation.java

**public** **class** My\_Calculation **extends** Task31 {

**public** **void** multiplication(**int** x, **int** y) {

z = x \* y;

System.***out***.println("The product of the given numbers: " + z);

}

**public** **static** **void** main(String args[]) {

**int** a = 20, b = 10;

My\_Calculation demo = **new** My\_Calculation();

demo.addition(a, b);

demo.Subtraction(a, b);

demo.multiplication(a, b);

}

}

Output:-  
The sum of the given numbers: 30

The difference between the given numbers: 10

The product of the given numbers: 200

**Task 33   
  
package** samplePackage8;

**class** Customer {

**int** cost = 40;

String items = "Tomatoes";

**void** purchase\_list() {

System.***out***.println("Purchased items: " + items);

System.***out***.println("Cost: " + cost);

}

}

**public** **class** Mart **extends** Customer {

**int** cost = 30;

String items = "Onions";

**void** billing() {

System.***out***.println("Billing items: " + items);

System.***out***.println("Billing cost: " + cost);

}

**public** **static** **void** main(String[] args) {

Mart m = **new** Mart();

// Accessing current class variables

System.***out***.println("Items: " + m.items);

System.***out***.println("Cost: " + m.cost);

// Accessing superclass variables using super

System.***out***.println("Super Items: " + ((Customer) m).items);

System.***out***.println("Super Cost: " + ((Customer) m).cost);

// Print symbols

System.***out***.println("%%%%%%%%%%%%%%");

// Method calls

m.purchase\_list();

m.billing();

}

}

Output:-  
Items: Onions

Cost: 30

Super Items: Tomatoes

Super Cost: 40

%%%%%%%%%%%%%%

Purchased items: Tomatoes

Cost: 40

Billing items: Onions

Billing cost: 30

**Task 34**

**package** samplePackage8;

**public** **class** Task34 {

**void** add(**int** x, **int** y) {

System.***out***.println("Two parameters: x = " + x + ", y = " + y);

}

**void** add(**int** x, **int** y, **int** z) {

System.***out***.println("Three parameters: x = " + x + ", y = " + y + ", z = " + z);

}

**public** **static** **void** main(String[] args) {

Task34 obj = **new** Task34();

obj.add(10, 20);

obj.add(5, 15, 25);

}

}

Output:-  
Two parameters: x = 10, y = 20

Three parameters: x = 5, y = 15, z = 25

**Task 35**  
**package** samplePackage8;

**public** **class** Task35 {

**void** add(**char** x, **char** y) {

System.***out***.println("Char values: x = " + x + ", y = " + y);

}

**void** add(**int** x, **int** y) {

System.***out***.println("Int values: x = " + x + ", y = " + y);

}

**public** **static** **void** main(String[] args) {

Task35 obj = **new** Task35();

obj.add('d', 'a');

obj.add(100, 100);

}

}

Output:-

Char values: x = d, y = a

Int values: x = 100, y = 100

**Task 36**

**package** samplePackage8;

**public** **class** Task36 {

**void** add(**int** x, **float** y) {

System.***out***.println("Values: int x = " + x + ", float y = " + y);

}

**void** add(**float** x, **int** y) {

System.***out***.println("Values: float x = " + x + ", int y = " + y);

}

**public** **static** **void** main(String[] args) {

Task36 obj = **new** Task36();

obj.add(10.50f, 60);

obj.add(100, 80.80f);

}

}

Output:-  
Values: float x = 10.5, int y = 60

Values: int x = 100, float y = 80.8

**Task 37  
  
package** samplePackage8;

**class** Employee {

**private** **int** pwd; // Private: accessed via getter/setter

**protected** **int** salary; // Protected: accessible in subclass

**public** **int** empid; // Public: directly accessible

// Default constructor

Employee() {

System.***out***.println("Employee constructor called");

}

// Setter for private variable pwd

**public** **void** setPwd(**int** pwd) {

**this**.pwd = pwd;

}

// Getter for private variable pwd

**public** **int** getPwd() {

**return** pwd;

}

// Method to display values

**public** **void** showDetails() {

System.***out***.println("Password: " + pwd);

System.***out***.println("Salary: " + salary);

System.***out***.println("EmpID: " + empid);

}

}

**class** Hr **extends** Employee {

Hr() {

**super**(); // Call parent constructor

setPwd(1254);

salary = 50000

empid = 10001

}

**public** **void** showHrDetails() {

showDetails(); // Display all info

}

**public** **static** **void** main(String[] args) {

Hr hr = **new** Hr();

hr.showHrDetails();

}

}

Output:-  
Employee constructor called

Password: 1254

Salary: 50000

EmpID: 10001

**Task 38**  
Public class AbstractDemo {

   public static void main(String [] args) {

      /\* Following is not allowed and would raise error \*/

      Employee e = new Employee("George W.", "Houston, TX", 43);

      System.out.println("\n Call mailCheck using Employee reference--");

      e.mailCheck();

   }

}

abstract class Employee {

   private String name;

   private String address;

   private int number;

   public Employee(String name, String address, int number) {

      System.out.println("Constructing an Employee");

      this.name = name;

      this.address = address;

      this.number = number;

   }

   public double computePay() {

     System.out.println("Inside Employee computePay");

     return 0.0;

   }

   public void mailCheck() {

      System.out.println("Mailing a check to " + this.name + " " + this.address);

   }

   public String toString() {

      return name + " " + address + " " + number;

   }

   public String getName() {

      return name;

   }

   public String getAddress() {

      return address;

   }

   public void setAddress(String newAddress) {

      address = newAddress;

   }

   public int getNumber() {

      return number;

   }

}

Output:-  
  
Employee e = new Employee("George W.", "Houston, TX", 43); // ❌ Error: Cannot instantiate abstract class  
  
  
  
**Task 39  
package** samplePackage8;

**public** **class** AbstractDemo {

**public** **static** **void** main(String[] args) {

PermanentEmployee e = **new** PermanentEmployee(

"Ranga Srinivas",

"Hyderabad, Telangana",

43,

"ranga.srinivas@example.com",

"+91-9876543222210"

);

System.***out***.println("\nCall mailCheck using Employee reference--");

e.mailCheck();

e.showEmployeeDetails();

System.***out***.println("Computed Pay: $" + e.computePay());

}

}

**abstract** **class** BaseEmployee {

**private** String name;

**private** String address;

**private** **int** number;

**public** BaseEmployee(String name, String address, **int** number) {

System.***out***.println("Constructing an Employee");

**this**.name = name;

**this**.address = address;

**this**.number = number;

}

**public** **abstract** **double** computePay();

**public** **void** mailCheck() {

System.***out***.println("Mailing a check to " + **this**.name + ", " + **this**.address);

}

**public** String toString() {

**return** name + " " + address + " " + number;

}

**public** String getName() {

**return** name;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String newAddress) {

address = newAddress;

}

**public** **int** getNumber() {

**return** number;

}

}

**class** PermanentEmployee **extends** BaseEmployee {

**private** String designation = "Associate";

**private** String department = "CXQO";

**private** String email;

**private** String phoneNumber;

**public** PermanentEmployee(String name, String address, **int** number, String email, String phoneNumber) {

**super**(name, address, number);

**this**.email = email;

**this**.phoneNumber = phoneNumber;

}

**public** **double** computePay() {

System.***out***.println("Inside PermanentEmployee computePay");

**return** 4000.0;

}

**public** **void** showEmployeeDetails() {

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

System.***out***.println("Address: " + getAddress());

System.***out***.println("Employee Number: " + getNumber());

System.***out***.println("Designation: " + designation);

System.***out***.println("Department: " + department);

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

System.***out***.println("Phone Number: " + phoneNumber);

}

}

Output:-

Constructing an Employee

Call mailCheck using Employee reference--

Mailing a check to Ranga Srinivas, Hyderabad, Telangana

Name: Ranga Srinivas

Address: Hyderabad, Telangana

Employee Number: 43

Designation: Associate

Department: CXQO

Email: ranga.srinivas@example.com

Phone Number: +91-9876543222210

Inside PermanentEmployee computePay

Computed Pay: $4000.0

**Task 40**abstract class Gadgets {

abstract void turnOn();

abstract void turnOff();

}

class TVRemote extends Gadgets {

@Override

void turnOn() {

System.out.println("TV is turned ON.");

}

@Override

void turnOff() {

System.out.println("TV is turned OFF.");

}

}

class ACRemote extends Gadgets {

@Override

void turnOn() {

System.out.println("AC is turned ON.");

}

@Override

void turnOff() {

System.out.println("AC is turned OFF.");

}

}

// Added missing classes

class FanRemote extends Gadgets {

@Override

void turnOn() {

System.out.println("Fan is turned ON.");

}

@Override

void turnOff() {

System.out.println("Fan is turned OFF.");

}

}

class CoolerRemote extends Gadgets {

@Override

void turnOn() {

System.out.println("Cooler is turned ON.");

}

@Override

void turnOff() {

System.out.println("Cooler is turned OFF.");

}

}

public class Main {

public static void main(String[] args) {

Gadgets remote;

remote = new TVRemote();

remote.turnOn();

remote.turnOff();

remote = new ACRemote();

remote.turnOn();

remote.turnOff();

remote = new FanRemote();

remote.turnOn();

remote.turnOff();

remote = new CoolerRemote();

remote.turnOn();

remote.turnOff();

}

}  
output:-   
TV is turned ON.

TV is turned OFF.

AC is turned ON.

AC is turned OFF.

Fan is turned ON.

Fan is turned OFF.

Cooler is turned ON.

Cooler is turned OFF.   
  
  
**Task 41**   
import java.io.\*;

// Interface Declared

interface testInterface {

// public, static and final

final int a = 10;

// public and abstract

void display();

}

// Class implementing interface

class TestClass implements testInterface {

// Implementing the capabilities of Interface

public void display(){

System.out.println("Myclass");

}

}

class Myclass

{

public static void main(String[] args)

{

TestClass t = new TestClass();

t.display();

System.out.println(t.a);

}

}  
  
output:-   
  
Myclass

10