

WEEK 1 Data types, operators

Skill builder:-

- 1) Input : Output :
- 90 The integer closer to 100 is 90 with a
80 difference of 10.

CODE:

```
import java.util.Scanner;  
class Main {  
    public static void main (String args []) {  
        Scanner sc = new Scanner (System.in);  
        int n = sc.nextInt();  
        int m = sc.nextInt();  
        int c = Math.abs (100-n);  
        int d = Math.abs (100-m);  
        if (c < d)  
            System.out.println ("the Integer closer to  
100 is "+n+" with a difference of "+c);  
        else  
            System.out.println ("the Integer closer to  
100 is "+m+" with a difference of "+d);  
    }  
}
```

- 2) Input : Output :
- 4 One of the integers is positive
3 while the other is not divisible by 3.

CODE:

```
import java.util.Scanner;  
class Main {  
    public static void main (String args []) {  
    }
```

```

{ scanner ob = new Scanner (System.in);
  int a = ob.nextInt();
  int b = ob.nextInt();
  if (((a%3) + (b%3) == 0) || ((b%3) + (a%3) == 0)){
    System.out.println ("One of the integers is positive
while the other is not divisible by 3.");
  } else {
    System.out.println ("Neither of the integers meets the
condition.");
  }
}

```

3)

Input:

20

Output

Original Integer : 20

Converted Double : 20.0

CODE:

```

import java.util.Scanner;
class Main {
  public static void main (String args[]){
    Scanner obj = new Scanner (System.in);
    int a = obj.nextInt();
    double b = a;
    System.out.println ("original Integer :" + a);
    System.out.println ("converted Double :" + b);
  }
}

```

4)

Input

2 2

Output

Sum is multiple of Product

CODE:

```

import java.util.Scanner;
class Main {
  public static void main (String [ ] args){
}

```

```

Scanner obj = new Scanner (System.in);
int n = obj.nextInt();
int m = obj.nextInt();
int d = n+m;
int c = n*m;
if ((c+d) == 0)
    System.out.println ("Sum is not Multiple of Product");
else
    System.out.println ("Sum is Multiple of product");
}

```

5) Input

3.0

Output:

Circumference: 18.85 meters

Area: 28.27 square meters

CODE:

```

import java.util.Scanner;
class Main {
    public static void main (String args[]) {
        Scanner ob = new Scanner (System.in);
        double r = ob.nextDouble();
        double cir = 2 * 3.14159 * r;
        double area = 3.14159 * r * r;
        String a = String.format ("% .2f", cir);
        String b = String.format ("% .2f", area);
        System.out.println ("Circumference : " + a + " meters");
        System.out.println ("Area: " + b + " square meters");
    }
}

```

6) Input

85

Output:

2

Result : 1

CODE:

```
import java.util.Scanner;
class Main {
    public static void main (String args) {
        Scanner sc = new Scanner (System.in);
        int x = sc.nextInt();
        int N = sc.nextInt();
        int a = (1 << N) - 1;
        int b = x + a;
        System.out.println ("result : " + b);
    }
}
```

7) Input:

15

Output:

false

345

CODE:

```
import java.util.*;
class Main {
    public static void main (String args[]) {
        Scanner obj = new Scanner (System.in);
        int a = obj.nextInt();
        int b = obj.nextInt();
        boolean res = ((a <= 0 && a >= 0) || (b <= 0 && b >= 0));
        System.out.println (res);
    }
}
```

8)

Input:

5

Output:

38.33

100

Average is greater than both 5 and 100.

10

CODE:

```
import java.util.*;
```

Class Main {

```
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int f = sc.nextInt();
        int c = sc.nextInt();
        int u = sc.nextInt();
        float av = (float) (f+c+u)/3;
        System.out.printf ("%-.2f\n", av);
        if ((av>f) && (av>c) && (av>u))
            System.out.println ("Average is greater than
both " + f + " and " + c);
        else if ((av>c) && (av>u))
            System.out.println ("Average is greater than
both " + c + " and " + u);
        else if ((av>u) && (av>f))
            System.out.println ("Average is greater than
both " + f + " and " + u);
        else
            System.out.println ("Average is not greater
than two smallest expenses");
    }
}
```

Q) Input:

1
5
9

Output :

3.33

true

(the second integer is halfway

between the first and third integer)

Code : // Program to check if the second integer

```
import java.util.Scanner;
```

Class Main {

```
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int s1 = sc.nextInt();
        int s2 = sc.nextInt();
        int s3 = sc.nextInt();
    }
}
```

```

int mid = (s1+s2)/2;
if (mid == s2) {
    System.out.println (true);
    System.out.println ("The second integer is halfway between
first and third integers.");
} else {
    System.out.println (false);
    System.out.println ("The second integer is not halfway
between first and third integers.");
}

```

3

Output:

- (a) Input: "Workdone in first d1 days (A+B+C): 0.54
8 12 16 21 Workdone in next d2 days (A+B): 0.24
Remaining work : 0.25
("Work" has 3 lines)

CODE:

```

import java.util.Scanner;
class Main {
    public static void main (String args []) {
        Scanner ob = new Scanner (System.in);
        String str = ob.nextLine();
        int A = ob.nextInt();
        int B = ob.nextInt();
        int C = ob.nextInt();
        int d1 = ob.nextInt();
        int d2 = ob.nextInt();
        double d1days = ((1.0/A)+(1.0/B)+(1.0/C))*d1;
        double d2days = ((1.0/A)+(1.0/B))*d2;
        double rem = 1-(d1days+d2days);
        System.out.printf ("Workdone in first d1 days (A+B+C): %f\n", d1days);
        System.out.printf ("Workdone in next d2 days (A+B): %f\n", d2days);
        System.out.printf ("Remaining Work : %f", rem);
    }
}

```

WEEK 2: CONTROL STRUCTURES

SKILL BUILDER:

1) INPUT:

39 25 30 45 67

Output:

Average score: 41
The student has failed

CODE:

```

import java.util.Scanner;
class Main {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int m1 = sc.nextInt();
        int m2 = sc.nextInt();
        int m3 = sc.nextInt();
        int m4 = sc.nextInt();
        int m5 = sc.nextInt();
        int avg = (m1+m2+m3+m4+m5)/5;
        System.out.println ("Average score:" + avg);
        if (avg >= 50)
            System.out.println ("The student has passed");
        else
            System.out.println ("The student has failed");
    }
}
  
```

2) Input:

37

Output

37 is neither multiple of 5 nor 7

CODE:

```

import java.util.Scanner;
class Main {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int a = sc.nextInt();
        if (a%5==0 || a%7==0)
            System.out.println (a + " is a multiple of 5");
    }
}
  
```



```

int ic = sc.nextInt();
int y = sc.nextInt();
double cr = ic;
double dp = 0.15;
for (int i=1; i<=y; i++) {
    cr = cr + (1-dp);
    System.out.println("current value : " + cr);
    if (cr > 10000)
        System.out.println("category : High");
    else if (cr >= 5000 && cr <= 10000)
        System.out.println("category : Medium");
    else
        System.out.println("category : Low");
}

```

- 5) Input : 20 Output :
 the no: of digits in 20 matches the sum of
 its digits.

CODE:

```

import java.util.Scanner;
public class Main
{
    public static void main (String [ ] args)
    {
        Scanner sc = new Scanner (System.in);
        int num = sc.nextInt();
        int sum = 0, c = 0;
        int temp = num;
        while (temp != 0)
        {
            sum += temp % 10;
            temp = temp / 10;
            c++;
        }
        if (sum == c)
            System.out.println ("The no: of digits in " + num +
                " matches the sum of its digits");
        else
            System.out.println ("The no: of digits in " + num +
                " does not match sum of its digits");
    }
}

```

6) INPUT

5

Output

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

CODE:

```
import java.util.Scanner;  
  
public class Main  
{  
    public static void main (String [] args)  
    {  
        Scanner sc = new Scanner (System.in);  
        int n = sc.nextInt();  
  
        for (int i=1; i<=n; i++)  
        {  
            for (int j=1; j<=i; j++)  
            {  
                System.out.print ("* ");  
            }  
            System.out.println ();  
        }  
        for (int i=n-1; i>=1; i--)  
        {  
            for (int j=1; j<=i; j++)  
            {  
                System.out.print ("* ");  
            }  
            System.out.println ();  
        }  
    }  
}
```

7) INPUT:

4

Output:

1
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6

CODE:

```
import java.util.Scanner;  
  
class Main  
{  
    public static void main (String [] args)  
    {  
        Scanner sc = new Scanner (System.in);  
        int n = sc.nextInt();  
    }  
}
```

```

for (int i=1; i<=n; i++) {
    for (int j=i; j<=n; j++) {
        System.out.print(" ");
    }
    for (int j=1; j<=(2*i-1); j++) {
        System.out.print(j);
    }
    System.out.println();
}

```

8) **Input:**

3

Output:

102
105
108

```

class Main {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int N = sc.nextInt();
        int c=0;
        for (int i=1; i<=N; i++) {
            for (int j=0; j<=i; j++) {
                for (int k=0; k<=j; k++) {
                    if (i!=j && j!=k && i!=k) {
                        int num = i * 100 + j * 10 + k;
                        if (num % 3 == 0) {
                            System.out.println (num);
                            c++;
                        }
                        if (c==N) break;
                    }
                    if (c==N) break;
                }
                if (c==N) break;
            }
        }
    }
}

```

WEEK 3: ARRAYS

Q1) Input & Output

1) Input

3

10 28 47

Output:

38

CODE:

```
import java.util.Arrays;
import java.util.Scanner;
class Main {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int n = sc.nextInt();
        int [] arr = new int [n];
        for (int i=0; i<n; i++) {
            arr[i] = sc.nextInt();
        }
        Arrays.sort (arr);
        int small = arr[0];
        int large = arr [arr.length - 1];
        int sum = large + small;
        System.out.println (sum);
    }
}
```

2)

Input

3

1 2 5

3 6 4

8 5 3

Output

Sum of main diagonal : 10

Sum of secondary diagonal : 19

CODE:

```
class Main {
```

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

```
        int main=0, sec=0;
```

```
        Scanner sc = new Scanner (System.in);
```

```
        int n = sc.nextInt();
```

```
        int [][] mat = new int [n][n];
```

```

for (int i=0; i<n-1; i++) {
    for (int j=0; j<n; j++) {
        mat[i][j] = sc.nextInt();
    }
}

for (int i=0; i<n-1; i++) {
    main += mat[i][i];
    sec += mat[i][n-1-i];
}

System.out.println("sum of main diagonal " + main);
System.out.println("sum of secondary diagonal " + sec);

```

1

2) Input:

5

10 20 30 40 50

Output:

Sum of first and last element : 60

CODE:

```

class Main {
    public static void main (String [] args) {
        Scanner s = new Scanner (System.in);
        int n = s.nextInt();
        int [] arr = new int [n];
        for (int i=0; i<n; i++) {
            arr[i] = s.nextInt();
        }
        int a = arr[0] + arr[n-1];
        System.out.println ("sum of first and last elements " + a);
    }
}

```

4) Input

1 3

25 5 5

5 44 5

Output

30 49 10

Class A Main {

```
public static void main (String [] args) {
    Scanner sc = new Scanner (System.in);
    int r = sc.nextInt();
    int c = sc.nextInt();
    int m1 [][] = new int [r][c];
    int m2 [][] = new int [r][c];
    int sum [][] = new int [r][c];
    for (int i=0; i<r; i++) {
        for (int j=0; j<c; j++) {
            m1 [i][j] = sc.nextInt();
        }
    }
    for (int i=0; i<r; i++) {
        for (int j=0; j<c; j++) {
            m2 [i][j] = sc.nextInt();
        }
    }
    for (int i=0; i<r; i++) {
        for (int j=0; j<c; j++) {
            sum [i][j] = m1 [i][j] + m2 [i][j];
            System.out.print (sum [i][j] + " ");
        }
        System.out.println ();
    }
}
```

5)

Input

5
15 23 15 34 23

Output

Class Main {

```
public static void main (String [] args) {
    Scanner sc = new Scanner (System.in);
    int n = sc.nextInt();
    int arr [] = new int [n];
    for (int i=0; i<n; i++) {
        arr [i] = sc.nextInt();
    }
}
```

```

Arrays. sort (arr);
boolean dup = false;
for (int i=0; i<n-1; i++) {
    if (arr[i] == arr[i+1]) {
        System.out.println (arr[i]);
        dup = true;
        break;
    }
}
if (!dup)
    System.out.println ("No repeated");

```

WEEK 4: STRING:

SKILL BUILDER:

D Input

Hello, world. How are you?

Output

1 1 1

CODE:

```

class Main {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int t = sc.nextInt();
        for (int i=0; i<n; i++) {
            String arr = sc.nextLine();
            int c=0, p=0, q=0;
            for (int j=0; j<arr.length(); j++) {
                char ch = arr.charAt (j);
                if (ch == 'c') c++;
                else if (ch == '.') p++;
                else if (ch == '?') q++;
            }
            System.out.println (c + " " + p + " " + q);
        }
    }
}

```

2) Input

5
Blockchain Cloud AI Data
cyber security

Output

AI Blockchain Cloud
Cyber Security Data

CODE:

class Main{

```
public static void main (String [] args){  
Scanner sc = new Scanner (System.in);  
int n = Integer.parseInt (sc.nextLine());  
String [] keywords = sc.nextLine().split (" ");  
Arrays.sort (keywords);  
for (int i=0; i<n; i++) {  
System.out.println (keywords [i]);  
if (i!=n-1) System.out.print (" ");  
}  
}
```

3)

Input

1
8939425917

Output

YES

CODE:

class Main {

```
public static void main (String [] args){  
Scanner sc = new Scanner (System.in);  
int T = sc.nextInt();  
for (int i=0; i<T; i++) {  
String s = sc.nextLine().trim();  
if (s.length () == 10 && s.matches ("\\d{10}")) {  
if (s.charAt (0) != '0') {  
System.out.println ("YES");  
} else {  
System.out.println ("No");  
}  
}  
}
```

4) Input

a b c e

Output

No valid words.

CODE: class Main {

public static void main (String [] args) {

Scanner sc = new Scanner (System.in);

String sentence = sc.nextLine();

String [] words = sentence.split (" ");

StringBuilder validwords = new StringBuilder();

for (String word : words) {

if (word.length() >= 2 & word.matches

("[A-Z][a-z]*")) {

validwords.append (word).append (" ");

}

} if (validwords.length() > 0) {

System.out.print (validwords.toString().trim());

}

else {

System.out.print ("No valid words");

}

sc.close();

5) Input:

1 2 3 4 5 6 7 8 9 0

12345

Output

No valid words

CODE: private static boolean isValid (String pin) {

If (pin.length () != 4) { return false; }

for (int i = 0; i < 4; i++) {

if (!Character.isDigit (pin.charAt (i)))

return false; }

```

char firstCh = pin.charAt(0);
for (int i=1; i<4; i++) {
    if (pin.charAt(i) != firstCh) return true;
}
return false;

```

WEEK 5: CLASS & OBJECTS.

SKILL BUILDER:

D) Input:

8765

Patel

1500

3000

600

Output:

Customer Name: Patel

Final Bal: 2500.0

CODE:

```

import java.util.Scanner;

class Account {
    int accNo;
    String name;
    double bal;
    Account (int accNo, String name, double bal) {
        this.accNo = accNo;
        this.name = name;
        this.bal = bal;
    }
    void dep (double amt) {
        if (amt >= 0) bal += amt;
    }
    void withdraw (double amt) {
        if (amt <= bal) bal -= amt;
    }
}

class Bank {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int N = sc.nextInt();
        sc.nextLine();
    }
}

```

class Bank {

public static void main (String [] args) {

Scanner sc = new Scanner (System.in);

int N = sc.nextInt();

sc.nextLine();

```

for (int i=0; i<N; i++) {
    int accNo = sc.nextInt();
    String name = sc.nextLine();
    double initBal = sc.nextDouble();
    double dep = sc.nextDouble();
    double with = sc.nextDouble();
    Account a = new Account (accNo, name, initBal);
    a.deposit (dep);
    a.withdraw (with);
    System.out.println ("Customer name: " + a.name);
    System.out.printf ("Final Bal : %.1f\n", a.bal);
}

```

2)

Input

1002
Sharma
103

Output

ID: 1002
Name: Sharma
Bill : 4521.0

CODE:

```

import java.util.Scanner;
class Customer{
    int id;
    String name;
    double units;
    Customer (int id, String name, double units) {
        this.id = id;
        this.name = name;
        this.units = units;
    }
    double calculateBill () {
        double bill = 0;
        if (units <=100)
            bill = units * 5;
        else if (units <=200)
            bill = 100 * 5 + (units - 100) * 7;
        else
            bill = 100 * 5 + 100 * 7 + (units - 200) * 9;
    }
}

```

```
if (bill > 2000) bill = bill * 0.95;
```

```
return bill;
```

```
}
```

```
class Main
```

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

```
Scanner sc = new Scanner (System.in);
```

```
int N = Integer.parseInt (sc.nextLine ());
```

```
for (int i=0 ; i<N ; i++) {
```

```
int id = sc.nextInt ();
```

```
String name = sc.nextLine ();
```

```
double units = sc.nextDouble ();
```

```
Customer c = new Customer (id, name, units);
```

```
System.out.println ("ID " + c.id);
```

```
System.out.println ("name " + c.name);
```

```
System.out.println ("Bill : " + c.calculateBill());
```

3) Input:

1

1234

Rahul

15

Output

ID : 1234

Name: Rahul

Fare : 200.0

CODE:

```
import java.util.Scanner;
```

```
class Booking {
```

```
private int bookingID;
```

```
private String cusName;
```

```
private double distance;
```

```
public Booking (int bookingID, String cusName,  
double distance) {
```

```
this.bookingID = bookingID;
```

```
this.cusName = cusName;
```

```
this.distance = distance; }
```

```

public void setBookingID (int bookingID) {
    this.bookingID = bookingID;
}

public void setCustomerName (String cusName) {
    this.cusName = cusName;
}

public void setDistance (double distance) {
    this.distance = distance;
}

public int getBookingID() {
    return bookingID;
}

public String getCustomerName () {
    return cusName;
}

public double getDistance () {
    return distance;
}

public double calculate() {
    double baseFare = 250;
    double perKmcharge = 10;
    double totalFare = baseFare + (distance * perKmcharge);
    if (distance > 20) {
        totalFare = totalFare * 0.9;
    }
    return totalFare;
}

```

A)

Input

1
1234
Mani
₹

JD: 1234

name: Mani

Final Fee: 3400.00

CODE:

```

import java.util.Scanner;
class student {
    private int empID;
    private String studName;
    private int numSub;
}

```

```

public student (int enrollmentId, String studName, int
numsub) {
    this.enrollmentId = enrollmentId;
    this.studName = studName;
    this.numsub = numsub;
}

public void setEnrollmentId (int enrollmentId) {
    this.enrollmentId = enrollmentId;
}

public void setstudName (String studName) {
    this.studName = studName;
}

public int getEnrollmentId () {
    return enrollmentId;
}

String getstudName () {
    return studName;
}

double calculateFee () {
    double registrationFee = 1000;
    double subjectFee = numsub * 800;
    double total = registrationFee + subjectFee;

    if (numsub > 5) {
        totalFee = totalFee - (0.20 * totalFee);
    }
    return totalFee;
}

```

WEEK 6: INHERITANCE:-

- 1) Program to calculate the total monthly cost.

Input

98.56

99.67

94.56

Output

Rs. 292.79

CODE :

Class Subscription

```
double baseMonCost;
```

double Ser Tax;

```
double GetTax(),  
subscription (double baseMonCost, double servTax) {
```

thus. $\text{basef}on\text{cost}$ = $\text{baseMon}\text{cost}$

this, serviceTax = serviceTax;] }

Class PreemptiveSubscription extends subscription E

double extraFeatureCost;

Penicill Subscription (double basePenCost, double seatTax,
double extraFeatureCost) {

~~super (baseMonthlyCst , ServiceTax);~~

```
this.extraFeatureCust = extraFeatureList;}
```

decorator calculateMonthly(result) {

return basection cost + Service Tax + extraFeature cost;

return based on

2) calculate the selling price of product after applying discount

Inhalt

Output

50:00

Rs. 40.00

0-20

COPE : Class Product

public double price;

Product (double price)?

this price = price: 3

class Discount extends Product {

private double disRate;

Dissent (double price, double dispute)?

super (palce);

~~temp~~ disRate = disRate;

```
double calculateSelling() {
```

```
    if(disRate > 1) { return -1; }
```

```
    return price * (1 - disRate);
```

```
}
```

- ③ calculate the final price of the item after applying tax:

Input

200

15

205.0

8.5

Output

230

222.43

CODE:

```
class SaleTaxCal {
```

```
public static int calculateFinal (int price, int taxRate) {
```

```
    return price + (price * taxRate) / 100; }
```

```
public static double calculateFinal (double price,  
                                     double taxRate) {
```

```
    return price + (price * taxRate) / 100.0; }
```

```
}
```

- ④ Display the calculated volumes of cube & cuboid

Input

12.1 13.4 17.6

23.7

Output

Vol of cuboid : 2853.66

Vol of cube : 13312.05

CODE:

```
class cuboid {
```

```
    double length, width, height;
```

```
    cuboid (double length, double width, double height)
```

```

    this.length = length;
    this.width = width;
    this.height = height; }

    double calculateVolume() {
        return length * width * height; }

}

class Cube extends Cuboid {
    double side;
    Cube(double side) : super(side, side, side) {
        this.side = side; }

    double calculateVolume() { return side * side * side; }

}

```

Display the total cost + shopping list after applying disc.

⑤

Input

1

Reg banana 1.99

Output

1.99

```

CODE: class Item {
    protected String name; protected double price;
    public Item (String name, double price) {
        this.name = name; this.price = price; }

    public double calculateCost() { return price; }

}

class Produce extends Item {
    public Produce (String name, double price) {
        super(name, price); }

    public double calculateCost() { return price; }

}

class Organic extends Produce {
    public Organic (String name, double price) {
        super(name, price); }

    public double calculateCost() { return price * 0.9; }

}

```

WEEK 7: INTERFACE.

calculate the total cost based on the daily energy usage

1) Input

100.0

1

1000.0

Output

Daily-energy cost:

Day 1: Rs. 100000.00

Total : Rs. 100000.00

CODE:

```
interface CostCalculator {  
    void get (Scanner scanner);  
    void display();  
}
```

Class Tracker implements CostCalculator {

private double rate;

private int days;

private double [] energyCons;

public void get (Scanner scanner) {

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

energyCons[i] = scanner.nextDouble();

}

public void display () {

System.out.println ("Daily energy cost : ");

double totalCst = 0.0;

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

double dailyCst = energyCons[i] * rate;

totalCst += dailyCst;

System.out.printf ("Day %d : Rs. %.2f\n", i+1, dailyCst);

System.out.printf ("Total : Rs. %.2f\n", totalCst); } }

2) calculate BMI.

Input:

70.0

1.75

Output:

BMI : 22.86

CODE: interface calc { double calculateBMI (double weight, double height); }

class BMI implements calc {

public double calculateBMI (double weight, double height) {

if (weight <= 0 || height <= 0) { return -1; }

return weight / (height * height); }

3) calculate S.I.

INPUT:

1000.00
5.00
2

Output:

SI : 100.0

CODE:

```
interface Ical { double SI (double principal, double rate, int time); }
```

```
class SICalc implements Ical {
    public double SI (double principal, double rate, int time) {
        return (principal * rate * time) / 100;
    }
}
```

4) Find total inventory:

INPUT:

Laptop
800.0
3
2
5
3

Output:

product added
total : \$ 2400.0
Invalid choice

CODE:

```
class Product {
    private String name; private double price; private int quan;
    public Product (String name, double price, int quan) {
        this.name = name; this.price = price; this.quan = quan;
    }
    public double getvalue() {
        return price + quan;
    }
}
```

interface Inventory {

```
void addProd (String name, double price, int quan);
double calc(); }
```

class SInventory implements Inventory {

```
private double totalValue = 0.0;
public SInventory (int cap) { totalValue = 0.0; }
```

```
public void addProd (String name, double price, double quan) {
    double prodValue = price * quantity;
    totalValue += prodValue;
    System.out.println ("Product added");
}
```

```
public double calc() {
```

```
return totalValue; }
```

Q) calculate Age : Input 2000 Output 24 years old.

CODE :

Interface Age :

```
int calculateAge (int birthYear);
```

Class HumanAgeCalc implements Age

```
public int calculateAge (int birthYear) {
```

```
int CurrentYear = 2024;
```

```
return currentYear - birthYear; }
```

WEEKS EXCEPTION.

SKILL BOULDER.

Q) Program to validate the email :

Input:

Sample@gmail.com Valid

Output: Invalid

CODE

```
class DotException extends Exception {
```

```
public DotException (String msg) { super (msg); }
```

```
class AttheRateException extends Exception {
```

```
public AttheRateException (String msg) { super (msg); }
```

```
class DomainException extends Exception {
```

```
public DomainException (String msg) { super (msg); }
```

Public class Main {

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

```
Scanner sc = new Scanner (System.in);
```

```
String email = sc.nextLine(); }
```

```
public static void validateEmail (String email) throws
```

```
{ DotException, AtRateException, DomainException }
```

```

if (email.startsWith(".")) email.startsWith("@") ||
email.endsWith(".")) || email.endsWith("@"))
    throw new DotException ("Invalid");

int atIndex = email.indexOf('@');
String afterAt = email.substring(atIndex + 1);

String domain = parts[parts.length - 1];
List<String> validDomains = Arrays.asList("in", "com", "net", "biz");
if (!validDomains.contains(domain))
    throw new DomainException ("Invalid");

```

2) Input: 120 Output: Meeting scheduled successfully!

Code:

```

class InvalidDurationException extends Exception {
    public InvalidDurationException (String msg) { super(msg); }

class Scheduler {
    public static void validate (int n) throws InvalidDurationException {
        if (n <= 0 || n >= 240)
            throw new InvalidDurationException ("Invalid");
    }

    public static void main (String args) {
        try {
            validate (n);
            System.out.println ("Meeting scheduled successfully");
        } catch (InvalidDurationException e) {
            System.out.println ("Error: " + e.getMessage());
        }
    }
}

```

3) Username Validation: Input Alice Output Valid.

CODE: class InvalidUsernameException extends Exception {

public InvalidUsernameException (String msg) {super(msg);}

```

13 public class Main {
    public static void main (String args[]) {
        try {
            validateUsername ("username");
            System.out.println ("Username is Valid");
        } catch (InvalidUsernameException e) {
            System.out.print (e.getMessage ());
        }
    }
    public static void validateUsername (String username) throws InvalidUsernameException {
        if (username.contains (" ") || username.length () < 5) {
            System.out.println ("System.out.println");
            throw new InvalidUsernameException ("Invalid");
        }
    }
}

```

4) Find Eligible Voters: Input 20 Output Eligible.

CODE: class InvalidAgeException extends Exception {

public InvalidAgeException (String msg) {super(msg);}

```

public class Main {
    public static void main (String [] args) {
        try {
            validateAge (age);
            System.out.println ("Eligible");
        } catch (InvalidAgeException e) {
            System.out.println ("Error " + e.getMessage ());
        } catch (InputMismatchException e) {
            System.out.println ("Error " + e);
        }
    }
}

```

```
public static void validateAge (int age) throws  
    InvalidAgeException {  
    if (age < 18)  
        throw new InvalidAgeException ("Not Valid"); } }
```

5) Validate file Name: Input: Output:
 myfile123 Valid

```
Class InvalidFileNameException extends Exception {  
    public InvalidFileNameException (String msg) { super (msg); }}
```

```
}
```

```
public class Main { public static void main (String [ ] args) {
```

```
    try {  
        validateFileName (filename);  
        System.out.println ("Valid");  
    } catch (InvalidFileNameException e) {  
        System.out.println (e.getMessage());  
    }
```

```
    catch (Exception e)  
        System.out.println (e); } }
```

```
public static void validateFileName (String fileName)  
throws InvalidFileNameException {
```

```
    if (fileName.length () > 3 || !fileName.matches  
        ("[A-zA-Z0-9]+")) {
```

```
        throw new InvalidFileNameException ("Invalid"); } }
```

```
}
```