**Week 1**

## **Objective:** To understand the basic concepts of Object Oriented Programming System and to get familiar with object and class.

**Assignments:**

1. Write a Java program to print your name.

**package** Week1;

**public class** PrintName {

**public static void** main(String args[]){ System.***out***.println("UEMK");

}

}

**Output: UEMK**

## Write a Java program to add two numbers.

**package** Week1;

**public class** Sum {

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

**int** a=10; **int** b=20; **int** res=a+b;

System.***out***.println("Sum is = "+res);

}

}

**Output: 30**

## Write a Java program to change temperature from Celsius to Fahrenheit.

**package** Week1;

**public class** Celsius\_to\_Fahrenheit {

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

**float** temperature=(**float**) 37.7;

System.***out***.println("Temperature in Fahrenheit = " + temperature); temperature = temperature \* 9/5 + 32;

System.***out***.println("Temperature in Celsius = " + temperature);

}

}

Temperature in Fahrenheit = 37.7 Temperature in Celsius = 99.86

1. Write a Java program to change temperature from Fahrenheit to Celsius.

**package** Week1;

**public class** Fahrenheit\_to\_Celsius {

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

**float** temperature=100;

System.***out***.println("Temperature in Fahrenheit = " + temperature); temperature = ((temperature - 32)\*5)/9;

System.***out***.println("Temperature in Celsius = " + temperature);

}

}

Output:

Temperature in Fahrenheit = 100.0 Temperature in Celsius = 37.77778

1. Write a Java program to find area and perimeter of a rectangle.

**package** Week1;

**public class** Celsius\_to\_Fahrenheit {

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

**float** length=10, width=10, area, perimeter;

perimeter = 2 \* (length + width); area = length \* width;

Output:

System.***out***.println("Perimeter of rectangle is " + perimeter + " units."); System.***out***.println("Area of rectangle is " + area + " sq. units.");

}

}

Perimeter of rectangle is 40.0 units. Area of rectangle is 100.0 sq. units.

1. Write a Java program to find area and perimeter of a circle.

**package** Week1;

**public class** Area\_Circle {

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

**double** radius=7.5;

**double** perimeter = 2 \* Math.***PI*** \* radius;

**double** area = Math.***PI*** \* radius \* radius;

System.***out***.println("Perimeter is = " + perimeter); System.***out***.println("Area is = " + area);

}

}

Perimeter is = 47.12388980384689 Area is = 176.71458676442586

1. Write a Java Program to display whether a number is odd or even.

**package** Week1;

**import** java.util.Scanner;

**public class** Odd\_Even {

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

Scanner reader = **new** Scanner(System.***in***); System.***out***.print("Enter a number: ");

**int** num = reader.nextInt();

**if**(num % 2 == 0) System.***out***.println(num + " is even");

**else**

System.***out***.println(num + " is odd");

}

}

Output:

Enter a number: 10 10 is even

1. Write a Java Program to check if a number is Positive or Negative.

**package** Week1; **import** java.util.Scanner; **public class** Pos\_Neg {

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

Scanner reader = **new** Scanner(System.***in***); System.***out***.print("Enter a number: ");

**int** num = reader.nextInt();

**if**(num < 0)

System.***out***.println(num + " is a negetive no.");

**else**

System.***out***.println(num + " is positive no.");

}

}

Output:

Enter a number: -999

-999 is a negetive no.

1. Write a Java program to find maximum of three numbers.

**package** Week1;

**import** java.util.Scanner;

**public class** Third\_Max {

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

**int** a=50; **int** b=30; **int** c=10;

**if**(a<b & a<c)

{

System.***out***.println("A is the third max");

}

**else if**(b<a & b<c)

{

}

**else**

{

}

System.***out***.println("B is the third max");

System.***out***.println("C is the third max");

}

}

Output: C is the third max

1. Write a Java program to swap two numbers.

**package** Week1;

**import** java.util.Scanner;

**public class** Swap {

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

**int** a=50; **int** b=30; **int** temp=a; a=b; b=temp;

System.***out***.println("A is "+a+" B is "+b);

}

}

Output: A is 30 B is 50

1. Write a Java program to convert miles to kilometers.

**package** Week1;

**import** java.util.Scanner;

**public class** Mile\_to\_Kilo {

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

**double** miles;

Scanner in = **new** Scanner(System.***in***);

System.***out***.println("Please enter miles:"); miles = in.nextDouble();

**double** kilometers = miles \* 1.6; System.***out***.println(kilometers + " Kilometers");

}

}

Output: Please enter miles:

10

16.0 Kilometers

1. Write a Java program to check whether a year is leap year or not.

**package** Week1;

**import** java.util.Scanner;

**public class** LeapYear {

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

**int** year = 1900;

**boolean** leap = **false**;

**if**(year % 4 == 0)

{

**if**( year % 100 == 0)

{

**if** ( year % 400 == 0) leap = **true**;

**else**

leap = **false**;

}

**else**

leap = **true**;

}

**else**

leap = **false**; **if**(leap)

System.***out***.println(year + " is a leap year.");

**else**

System.***out***.println(year + " is not a leap year.");

}

}

Output: 1900 is not a leap year.

1. Write a Java program for following grading system.

Note: Percentage>=90% : Grade A Percentage>=80% : Grade B Percentage>=70% : Grade C Percentage>=60% : Grade D Percentage>=40% :

Grade E Percentage<40% : Grade F

**package** Week1; **import** java.util.Scanner; **public class** Grade {

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

**float** avg;

Scanner scanner = **new** Scanner(System.***in***); System.***out***.print("Enter the %: "); avg=scanner.nextFloat(); System.***out***.print("The student Grade is: "); **if**(avg>=90)

{

System.***out***.print("A");

}

**else if**(avg>=80 && avg<90)

{

System.***out***.print("B");

}

**else if**(avg>=60 && avg<80)

{

System.***out***.print("C");

}

**else if**(avg>=40 && avg<60)

{

System.***out***.print("D");

}

**else**

{

System.***out***.print("E");

}

}

}

Output: Enter the %: 90 The student Grade is: A

1. Write a Java program to check whether a number is divisible by 5 or not.

**package** Week1; **import** java.util.Scanner; **public class** Mod\_Five {

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

**int** avg;

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the no.: "); avg=scanner.nextInt();

**if**(avg%5==0)

{

System.***out***.print("No. is divisible by 5 ");

}

**else**

{

System.***out***.print("No. is not divisible by 5 ");

}

}

}

Enter the no.: 25 No. is divisible by 5

**Week 2**

### Write a Java program to check whether a number is Buzz or not.

**public class** Buzz\_no {

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

**int** i = 70;

**if** (i%10==0 || i%7==0) System.***out***.println("Buzz Number");

### else

System.***out***.println("Not a Buzz Number");

}

}

Output: Buzz Number

### Write a Java program to calculate factorial of 12.

**public class** Factorial\_12 {

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

**int** i,sum=1;

**for**(i=1;i<=12;i++)

{

sum+=i\*sum;

}

System.***out***.println("Factorial of 12 is : "+sum);

}

}

Output: Factorial of 12 is : 1932053504

### Write a Java program for Fibonacci series.

**public class** Fibonacci {

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

**int** i = 1, n = 10, t1 = 0, t2 = 1; System.***out***.print("First " + n + " terms: "); **while** (i <= n)

{

**int** sum = t1 + t2; t1 = t2;

t2 = sum; i++;

}

}

}

Output: First 10 terms: 0 + 1 + 1 + 2 + 3 + 5 + 8 + 13 + 21 + 34

### Write a Java program to reverse a number.

**public class** Reverse {

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

**int** i =199,j,k=0;

**while** (i !=0)

{

j=i%10;

k=k\*10+j; i=i/10;

}

System.***out***.println("Answer is :"+k);

}

}

Output: Answer is :991

### Admission to a professional course is subject to the following conditions:

**(a) marks in Mathematics >= 60 (b) marks in Physics >=50**

**(c) marks in Chemistry >=40 (d) Total in all 3 subjects >=200 (Or)**

**Total in Maths & Physics>=150**

**Given the marks in the 3 subjects of n (user input) students, write a program to process the applications to list the eligible candidates.**

**import** java.util.Scanner;

**public class** Number\_Process {

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

**int** m,p,c;

Scanner sc=**new** Scanner(System.***in***); System.***out***.println("Enter marks of math :"); m=sc.nextInt();

System.***out***.println("Enter marks of phy :"); p=sc.nextInt();

System.***out***.println("Enter marks of chem :"); c=sc.nextInt();

**if**((m+p)>=150 || (p+c+m)>=200 )

{

}

### else

{

}

}

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

System.***out***.println("Not Eligable");

}

Output:

Enter marks of math :

60

Enter marks of phy :

95

Enter marks of chem :

45

Eligible

### Write a Java program to find all roots of a quadratic equation.

**import static** java.lang.Math.\*;

**public class** Root {

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

{

**int** a = 1, b = -7, c = 12;

**if** (a == 0)

{

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

### return;

}

**int** d = b\*b - 4\*a\*c;

**double** sqrt\_val = *sqrt*(*abs*(d)); System.***out***.println("Roots are real and different \n");

System.***out***.println((**double**)(-b + sqrt\_val) / (2 \* a) + "\n"

+ (**double**)(-b - sqrt\_val) / (2 \* a));

}

}

Roots are real and different

4.0

3.0

### Write a Java program to calculate the sum of natural numbers up to a certain range.

**public class** Root {

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

{

**int** sum=0;

**for**(**int** i=0;i<=10;i++)

{

sum+=i;

}

System.***out***.println("Sum is : "+sum);

}

}

Output: Sum is : 55

### Write a Java program to print all multiple of 10 between a given interval.

**import** java.util.Scanner;

**public class** Sum\_of\_Natural {

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

{

Scanner scanner=**new** Scanner(System.***in***); System.***out***.println("Enter the range : ");

**int** N = scanner.nextInt();

**for**(**int** i=1;i<=N;i++)

{

System.***out***.println(N+" x "+i+" = "+(10\*i));

}

}

}

Output: Enter the range :

10

10 x 1 = 10

10 x 2 = 20

10 x 3 = 30

10 x 4 = 40

10 x 5 = 50

10 x 6 = 60

10 x 7 = 70

10 x 8 = 80

10 x 9 = 90

10 x 10 = 100

### Write a Java program to generate multiplication table.

**import** java.util.Scanner;

**public class** Mult\_Table {

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

{

System.***out***.print("Enter the number: "); Scanner scanner=**new** Scanner(System.***in***);

**int** N = scanner.nextInt();

**for**(**int** i=1;i<=10;i++)

{

System.***out***.println(N+" x "+i+" = "+(N\*i));

}

}

}

OP: Enter the number: 5 5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

### Write a Java program to find HCF of two Numbers.

**import** java.util.Scanner;

**public class** HCF {

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

{

**int** n1 = 81, n2 = 153;

**while**(n1 != n2)

{

**if**(n1 > n2) n1 -= n2;

### else

n2 -= n1;

}

System.***out***.println("Answer is = " + n1);

}

}

Output: Answer is = 9

### Write a Java program to find LCM of two Numbers.

**import** java.util.Scanner;

**public class** LCM {

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

{

**int** n1 = 72, n2 = 120, gcd = 1;

**for**(**int** i = 1; i <= n1 && i <= n2; ++i)

{

**if**(n1 % i == 0 && n2 % i == 0) gcd = i;

}

**int** lcm = (n1 \* n2) / gcd;

System.***out***.printf("The LCM of %d and %d is %d.", n1, n2, lcm);

}

}

Output: The LCM of 72 and 120 is 360.

### Write a Java program to count the number of digits of an integer.

**public class** Count\_Digits {

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

**int** i =5555,counter=0;

**while** (i !=0)

{

i=i/10; counter++;

}

System.***out***.println("Count is :"+counter);

}

}

Output: Count is :4

### Write a Java program to calculate the exponential of a number.

**public class** Exponent\_Count {

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

**int** base = 3, exponent = 4;

**long** result = 1;

**while** (exponent != 0)

{

result \*= base;

--exponent;

}

System.***out***.println("Answer = " + result);

}

}

Answer = 81

### Write a Java program to check whether a number is palindrome or not.

**public class** Palindrome {

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

**int** num = 121, reversedInteger = 0, remainder, originalInteger; originalInteger = num;

**while**( num != 0 )

{

remainder = num % 10;

reversedInteger = reversedInteger \* 10 + remainder; num /= 10;

}

**if** (originalInteger == reversedInteger) System.***out***.println(originalInteger + " is a palindrome.");

### else

System.***out***.println(originalInteger + " is not a palindrome.");

}

}

Outout: 121 is a palindrome.

### Write a Java program to check whether a number is prime or not.

**public class** Prime {

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

**int** i,m=0,flag=0; **int** n=19; m=n/2; **if**(n==0||n==1){

System.***out***.println(n+" is not prime number");

}**else**{ **for**(i=2;i<=m;i++){ **if**(n%i==0){

System.***out***.println(n+" is not prime number"); flag=1;

### break;

}

}

**if**(flag==0) { System.***out***.println(n+" is prime number"); }

}

}

}

Output: 19 is prime number

### Write a Java program to convert a Binary Number to Decimal and Decimal to Binary.

**import** java.util.Scanner;

**public class** Binary\_Decimal {

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

**int** n, count = 0, a; String x = "";

Scanner s = **new** Scanner(System.***in***); System.***out***.print("Enter any decimal number:"); n = s.nextInt();

**while**(n > 0)

{

a = n % 2;

x = x + "" + a; n = n / 2;

}

System.***out***.println("Binary number:"+x);

**int** base = 1,dec\_value=0; System.***out***.print("Enter any Binary number:"); **int** num= s.nextInt();

**int** temp = num;

**while** (temp!=0) {

**int** last\_digit = temp % 10; temp = temp / 10;

dec\_value += last\_digit \* base;

base = base \* 2;

}

System.***out***.println("Decimal number:"+dec\_value);

}

}

Output:

Enter any decimal number:5 Binary number:101

Enter any Binary number:101 Decimal number:5

### Write a Java program to find median of a set of numbers.

**public class** Median {

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

**int** a[] = { 1,2,3,4,5,6,7,8};

**int** n = 8;

**if** (n % 2 != 0)

{

System.***out***.println("Median is = "+a[n / 2]);

}

### else

{

System.***out***.println("Median is = "+(a[(n - 1) / 2] + a[n / 2]) / 2.0);

}

}

}

Ouput: Median is = 4.5

### Write a program to compute the value of Euler’s number that is used as the base of natural logarithms. Use the following formula.

**e= 1+ 1/1! +1 /2! + 1/3+. 1/n!**

**public class** Eular\_Log {

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

**double** term = 1.0; **double** sum = 1.0; **int** n = 0;

**while** (term >= 0.0000001)

{

n++;

term = term/n; sum = sum + term;

}

System.***out***.println(" Approximate value of e is: "+sum);

}

}

Ouput: Approximate value of e is: 2.718281826198493

### Write a Java program to generate all combination of 1, 2, or 3 using loop.

**public class** Combination {

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

**int** i, j, k;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | **for** (i=1; i<=3; i++)  { |  | |
| **for** (j=1; j<=3; j++) |
| {  **for** (k=1; k<=3; k++) |
| System.***out***.println(i+"  } | "+j+" | "+k); |
| } |  |  |  | | |
| }}  1 | 1 | 1 |
| 1 | 1 | 2 |
| 1 | 1 | 3 |
| 1 | 2 | 1 |
| 1 | 2 | 2 |
| 1 | 2 | 3 |
| 1 | 3 | 1 |
| 1 | 3 | 2 |
| 1 | 3 | 3 |
| 2 | 1 | 1 |
| 2 | 1 | 2 |
| 2 | 1 | 3 |
| 2 | 2 | 1 |
| 2 | 2 | 2 |
| 2 | 2 | 3 |
| 2 | 3 | 1 |
| 2 | 3 | 2 |
| 2 | 3 | 3 |
| 3 | 1 | 1 |
| 3 | 1 | 2 |
| 3 | 1 | 3 |
| 3 | 2 | 1 |
| 3 | 2 | 2 |
| 3 | 2 | 3 |
| 3 | 3 | 1 |
| 3 | 3 | 2 |
| 3 | 3 | 3 |

### Write a Java program to read two integer values m and n and to decide and print whether m is multiple of n.

**import** java.util.Scanner;

**public class** Multiple\_Find {

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

{

System.***out***.print("Enter M: ");

Scanner scanner=**new** Scanner(System.***in***);

**int** M = scanner.nextInt(); System.***out***.print("Enter M: ");

**int** N = scanner.nextInt();

**if**(M%N==0)

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

### else

}

}

System.***out***.println("No");

Output: Enter M: 10 Enter M: 5

Yes

### Write a Java program to display prime numbers between a given interval.

**public class** Prime {

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

**int** i,m=0,flag=0; **int** n=19; m=n/2; **if**(n==0||n==1){

System.***out***.println(n+" is not prime number");

}**else**{ **for**(i=2;i<=m;i++){ **if**(n%i==0){

System.***out***.println(n+" is not prime number"); flag=1;

### break;

}

}

**if**(flag==0) { System.***out***.println(n+" is prime number"); }

}

}

}

Enter Up: 10

Enter Down: 20

11 is prime number 13 is prime number 17 is prime number 19 is prime number

### Write a Java program to check whether a given number is Armstrong Number or not.

**import** java.util.Scanner;

**public class** Amstrong {

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

**int** c=0,a,temp;

**int** n=153; temp=n; **while**(n>0)

{ a=n%10;

n=n/10; c=c+(a\*a\*a);

}

**if**(temp==c) System.***out***.println("armstrong number"); **else**

System.***out***.println("Not armstrong number");

}

}

armstrong number

### Write Java programs for the patterns given bellow: (23-25) 23.

1

2 3 4

5 6 7 8 9

**import** java.util.Scanner;

**public class** Pattern1 {

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

**int** i,j;

**for**(i=1;i<5;i++)

{

**for**(j=1;j<=i;j++)

{

System.***out***.print(j);

}

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

}

}

}

### 24.

1

2 3 2

3 4 5 4 3

4 5 6 7 6 5 4

**public class** Pattern3 {

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

**int** i, space, rows, k=0, count = 0, count1 = 0; rows=4;

**for**(i=1; i<=rows; ++i)

{

**for**(space=1; space <= rows-i; ++space)

{

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

++count;

}

**while**(k != 2\*i-1)

{

**if** (count <= rows-1)

{

System.***out***.print(i+k+" ");

++count;

}

### else

{

++count1;

System.***out***.print(i+k-2\*count1);

}

++k;

}

count1 = count = k = 0; System.***out***.println();

}

}

}

### 25.

1 1

2 2

3 3

4

**public class** Pattern2 {

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

**int** n=5;

**int** i, j;

**for** (i = n - 1; i >= 0; i--)

{

**for** (j = n - 1; j > i; j--)

{

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

}

System.***out***.print(i );

**for** (j = 1; j < (i \* 2); j++) System.***out***.print(" ");

**if** (i >= 1) System.***out***.print(i);

System.***out***.print("\n");

}

}

}

**Week 3**

1. Write a Java program to calculate Sum & Average of an integer array.

**import** java.util.Scanner;

**public class** Sum\_Avg\_Array {

**public static void** main(String args[]) { Scanner sc=**new** Scanner(System.***in***);

**int** arr[]=**new int**[10],n=10;

System.***out***.println("Enter 10 values");

**for**(**int** i=0;i<n;i++) { arr[i]=sc.nextInt();

}

**int** sum=0, avg;

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

{

sum+=arr[i];

}

avg=sum/n; System.***out***.println("Sum is "+sum); System.***out***.println("Avg is "+avg);

}

}

Enter 10 values

1

2

3

4

5

6

7

8

9

10

Sum is 55

Avg is 5

1. Write a Java program to implement stack using array.

**import** java.util.Scanner;

**public class** Stack {

**static final int** MAX = 1000;

**static int** top;

**int** a[] = **new int**[MAX];

**boolean** isEmpty()

{

**return** (top < 0);

}

Stack()

{

top = -1;

}

**boolean** push(**int** x)

{

**if** (top >= (MAX - 1)) {

System.out.println("Stack Overflow");

### return false;

}

### else {

a[++top] = x;

System.out.println(x + " pushed into stack");

### return true;

}

}

**int** pop()

{

**if** (top < 0) {

System.out.println("Stack Underflow");

**return** 0;

}

### else {

**int** x = a[top--];

**return** x;

}

}

**int** peek()

{

**if** (top < 0) {

System.out.println("Stack Underflow");

**return** 0;

}

### else {

**int** x = a[top];

**return** x;

}

}

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

{

Stack s = **new** Stack(); s.push(10);

System.out.println( "Peek of the stack "+s.peek()); s.push(20);

System.out.println( "Peek of the stack "+s.peek()); s.push(30);

System.out.println( "Peek of the stack "+s.peek()); System.out.println(s.pop() + " Popped from stack");

}

}

10 pushed into stack Peek of the stack 10 20 pushed into stack Peek of the stack 20 30 pushed into stack Peek of the stack 30 30 Popped from stack

1. Write a Java program to implement Queue using array.

**public class** Queue {

**int** front, rear, size;

**int** capacity;

**int** array[];

**public** Queue(**int** capacity) { **this**.capacity = capacity; front = **this**.size = 0;

rear = capacity - 1;

array = **new int**[**this**.capacity];

}

**boolean** isFull(Queue queue)

{ **return** (queue.size == queue.capacity);

}

**boolean** isEmpty(Queue queue)

{ **return** (queue.size == 0); }

**void** enqueue( **int** item)

{

**if** (isFull(**this**))

### return;

**this**.rear = (**this**.rear + 1)%**this**.capacity;

**this**.array[**this**.rear] = item;

**this**.size = **this**.size + 1; System.***out***.println(item+ " enqueued to queue");

}

**int** dequeue()

{

**if** (isEmpty(**this**))

**return** Integer.***MIN\_VALUE***;

**int** item = **this**.array[**this**.front];

**this**.front = (**this**.front + 1)%**this**.capacity;

**this**.size = **this**.size - 1;

**return** item;

}

**int** front()

{

**if** (isEmpty(**this**))

**return** Integer.***MIN\_VALUE***;

**return this**.array[**this**.front];

}

**int** rear()

{

**if** (isEmpty(**this**))

**return** Integer.***MIN\_VALUE***;

**return this**.array[**this**.rear];

}

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

{

Queue queue = **new** Queue(1000);

queue.enqueue(10); queue.enqueue(20); queue.enqueue(30); queue.enqueue(40);

System.***out***.println(queue.dequeue() + " dequeued from queue\n");

System.***out***.println("Front item is " +

queue.front());

System.***out***.println("Rear item is " +

queue.rear());

}

}

10 enqueued to queue 20 enqueued to queue 30 enqueued to queue 40 enqueued to queue 10 dequeued from queue

Front item is 20 Rear item is 40

1. Write a Java program to calculate Sum of two 2-dimensional arrays.

**import** java.util.Scanner;

**public class** Sum\_2D\_Array {

**public static void** main(String args[]) { Scanner s = **new** Scanner(System.***in***);

System.***out***.print("Enter number of rows: ");

**int** rows = s.nextInt();

System.***out***.print("Enter number of columns: ");

**int** columns = s.nextInt();

**int**[][] a = **new int**[rows][columns]; **int**[][] b = **new int**[rows][columns];

System.***out***.println("Enter the first matrix");

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

**for** (**int** j = 0; j < columns; j++) { a[i][j] = s.nextInt();

}

}

System.***out***.println("Enter the second matrix");

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

**for** (**int** j = 0; j < columns; j++) { b[i][j] = s.nextInt();

}

}

**int**[][] c = **new int**[rows][columns]; **for** (**int** i = 0; i < rows; i++) {

**for** (**int** j = 0; j < columns; j++) { c[i][j] = a[i][j] + b[i][j];

}

}

System.***out***.println("The sum of the two matrices is");

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

**for** (**int** j = 0; j < columns; j++) { System.***out***.print(c[i][j] + " ");

}

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

}

}

}

Enter number of rows: 2 Enter number of columns: 2 Enter the first matrix

1

2

3

4

Enter the second matrix 1

2

3

4

The sum of the two matrices is 2 4

6 8

1. Write a Java program to find the range of a 1D array.

**import** java.util.Arrays;

**public class** Array\_Range {

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

**int**[] arr = {2,4,2,4,5,6,3};

System.***out***.println("My array elements:\n");

**for**(**int** i=0;i<7;i++){ System.***out***.print(arr[i]+" ");

}

**int**[] newArr = Arrays.*copyOfRange*(arr, 1, 4); System.***out***.println("\nMy new array elements:\n"); **for**(**int** i=0;i<3;i++){

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

}

}

}

My array elements:

2 4 2 4 5 6 3

My new array elements:

4 2 4

1. Write a Java program to search an element in an array.

**public class** Linear\_search {

**public static int** search(**int** arr[], **int** x)

{

**int** n = arr.length;

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

{

**if**(arr[i] == x)

**return** i;

}

**return** -1;

}

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

{

**int** arr[] = { 2, 3, 4, 10, 40 };

**int** x = 10;

**int** n = arr.length,result=-1;

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

{

**if**(arr[i] == x) result=i;

}

**if**(result == -1)

System.***out***.print("Element is not present in array");

### else

System.***out***.print("Element is present at index " + result);

}

}

Element is present at index 3

1. Write a Java program to find the sum of even numbers in an integer array.

**import** java.util.Scanner;

**public class** Sum\_Even\_Array {

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

Scanner sc=**new** Scanner(System.***in***); **int** arr[]=**new int**[10],n=10; System.***out***.println("Enter 10 values"); **for**(**int** i=0;i<n;i++)

{

arr[i]=sc.nextInt();

}

**int** sum=0, avg;

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

{

**if**(arr[i]%2==0)

sum+=arr[i];

}

System.***out***.println("Sum of even ="+sum);

}

}

Enter 10 values

1

2

3

4

5

6

7

8

9

10

Sum of even =30

1. Write a Java program to find the sum of diagonal elements in a 2D array.

**public class** Sum\_of\_Dia {

**static void** printDiagonalSums(**int** [][]mat,

**int** n)

{

**int** principal = 0, secondary = 0;

**for** (**int** i = 0; i < n; i++) { principal += mat[i][i];

}

System.***out***.println(" Diagonal: "

+ principal);

}

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

{

**int** [][]a = { { 1, 2, 3, 4 },

{ 5, 6, 7, 8 },

{ 1, 2, 3, 4 },

{ 5, 6, 7, 8 } };

*printDiagonalSums*(a, 4);

}

}

Diagonal: 18

1. Reverse the elements in an array of integers without using a second array.

**public class** Reverse\_An\_Array {

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

{

**int** [] arr = {10, 20, 30, 40, 50};

**int** n=5,i=0,j=n-1;

**while**(i!=j)

{

**int** temp=arr[i]; arr[i]=arr[j]; arr[j]=temp; i++;

j--;

}

System.***out***.println("Reversed array is: \n");

**for** (**int** k = 0; k < n; k++) { System.***out***.println(arr[k]);

}

}

}

Output: Reversed array is:

50

40

30

20

10

1. Write a Java program to enter n elements in an array and find smallest number among them.

**public class** Reverse\_An\_Array {

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

{

**int** [] arr = {10, 20, -999, 40, 50};

**int** n=5,min=arr[0];

**for** (**int** k = 1; k < n; k++) {

**if**(min>arr[k])

min=arr[k];

}

System.***out***.println("Min element is "+min);

}

}

Min element is -999

1. Write Java program to find the sum of all odd numbers in a 2D array.

**public class** Sum\_of\_ODD\_Matrix {

**static void** printDiagonalSums(**int** [][]mat,

**int** n)

{

**int** sum=0;

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

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

**if**(mat[i][j]%2!=0)

sum+=mat[i][j];

}

}

System.***out***.println(" Sum of odd: "

+ sum);

}

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

{

**int** [][]a = { { 1, 2, 3, 4 },

{ 5, 6, 7, 8 },

{ 1, 2, 3, 4 },

{ 5, 6, 7, 8 } };

*printDiagonalSums*(a, 4);

}

}

Sum of odd: 32

1. Write a Java program to print transpose of matrix.

**public class** Transpose {

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

{

**int** [][]a = { { 1, 2, 3, 4 },

{ 5, 6, 7, 8 },

{ 1, 2, 3, 4 },

{ 5, 6, 7, 8 } };

**int** b[][]=**new int**[4][4],i,j; **for** (i = 0; i < 4; i++)

**for** (j = 0; j < 4; j++) b[i][j] = a[j][i];

System.***out***.print("Result matrix is \n");

**for** (i = 0; i < 4; i++)

{

**for** (j = 0; j < 4; j++) System.***out***.print(b[i][j] + " "); System.***out***.print("\n");

}

}

}

Result matrix is 1 5 1 5

2 6 2 6

3 7 3 7

4 8 4 8

1. Write a Java program to check whether a given matrix is sparse or not.

**import** java.util.Scanner;

**public class** Reverse\_An\_Array {

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

{

Scanner sc = **new** Scanner(System.***in***); System.***out***.println("Enter the dimensions of the matrix: "); **int** m = sc.nextInt();

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

**double**[][] mat = **new double**[m][n]; **int** zeros = 0;

System.***out***.println("Enter the elements of the matrix: ");

**for**(**int** i=0; i<m; i++)

{

**for**(**int** j=0; j<n; j++)

{

mat[i][j] = sc.nextDouble();

**if**(mat[i][j] == 0)

{

zeros++;

}

}

}

**if**(zeros > (m\*n)/2)

{

System.***out***.println("The matrix is a sparse matrix");

}

### else

{

System.***out***.println("The matrix is not a sparse matrix");

}

sc.close();

}

}

Enter the dimensions of the matrix:

2

2

Enter the elements of the matrix:

0

0

0

1

The matrix is a sparse matrix

1. Write a Java program to count the prime numbers in an array.

**public class** Prime\_Array {

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

**int** arr[]={1,2,3,4,5,6,7,8,9};

**for**(**int** k=0;k<9;k++){

**int** i,m=0,flag=0;

**int** n=arr[k]; m=n/2; **if**(n==0||n==1){

}**else**{ **for**(i=2;i<=m;i++){ **if**(n%i==0){ flag=1;

### break;

}

}

**if**(flag==0) { System.out.println(n+" is prime number"); }

}

}

}}

2 is prime number 3 is prime number 5 is prime number 7 is prime number

1. Write a Java program to find second highest element of an array.

**public class** Prime\_Array {

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

**int** arr[]={1,2,3,4,5,6,7,8,9};

**int** first,second;

first = second = arr[0];

**for** (**int** i = 0; i < 9; i++)

{

**if** (arr[i] > first)

{

second = first; first = arr[i];

}

**else if** (arr[i] > second && arr[i] != first) second = arr[i];

}

System.***out***.println("2nd is : "+second);

}}

2nd is : 8

1. Write a Java program which counts the non-zero elements in an integer array.

**public class** NonZero\_Count {

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

**int** arr[]={1,2,3,0,0,0,0,8,9};

**int** count=0;

**for** (**int** i = 0; i < 9; i++)

{

**if**(arr[i]!=0) count++;

}

System.***out***.println("Non zero element : "+count);

}}

Non zero element : 5

1. Write a Java program to merge two float arrays.

**public class** Merge\_Float {

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

**float**[]a = {1,2,3,4};

**float**[]b = {4,16,1,2,3,22};

**float**[]c = **new float**[a.length+b.length]; **int** count = 0;

**for**(**int** i = 0; i < a.length; i++) { c[i] = a[i];

count++;

}

**for**(**int** j = 0; j < b.length;j++) { c[count++] = b[j];

}

**for**(**int** i = 0;i < c.length;i++) System.***out***.print(c[i]+" ");

}

}

1.0 2.0 3.0 4.0 4.0 16.0 1.0 2.0 3.0 22.0

1. Write a Java program where elements of two integer arrays get added index wise and get stored into a third array.

**public class** Index\_wise\_add {

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

**int**[]a = {1,2,3,4};

**int**[]b = {4,16,1,2};

**int**[]c = **new int**[4]; **int** count = 0;

**for**(**int** i = 0; i < a.length; i++) { c[i]=a[i]+b[i];

}

5 18 4 6

}

**for**(**int** i = 0;i < c.length;i++) System.***out***.print(c[i]+" ");

}

1. Write a Java program to multiply two matrices.

**import** java.util.Scanner;

**public class** Multiply\_Matrix {

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

**int** m, n, p, q, sum = 0, c, d, k;

Scanner in = **new** Scanner(System.***in***);

System.***out***.println("Enter the number of rows and columns of first matrix"); m = in.nextInt();

n = in.nextInt();

**int** first[][] = **new int**[m][n]; System.***out***.println("Enter elements of first matrix");

**for** (c = 0; c < m; c++)

**for** (d = 0; d < n; d++) first[c][d] = in.nextInt();

System.***out***.println("Enter the number of rows and columns of second matrix"); p = in.nextInt();

q = in.nextInt();

**if** (n != p)

System.***out***.println("The matrices can't be multiplied with each other.");

### else

{

**int** second[][] = **new int**[p][q];

**int** multiply[][] = **new int**[m][q]; System.***out***.println("Enter elements of second matrix");

**for** (c = 0; c < p; c++)

**for** (d = 0; d < q; d++) second[c][d] = in.nextInt();

**for** (c = 0; c < m; c++)

{

**for** (d = 0; d < q; d++)

{

**for** (k = 0; k < p; k++)

{

sum = sum + first[c][k]\*second[k][d];

}

multiply[c][d] = sum; sum = 0;

}

}

System.***out***.println("Product of the matrices:");

**for** (c = 0; c < m; c++)

{

**for** (d = 0; d < q; d++) System.***out***.print(multiply[c][d]+"\t");

System.***out***.print("\n");

}

} }

}

Enter the number of rows and columns of first matrix 2

2

Enter elements of first matrix 10

20

30

40

Enter the number of rows and columns of second matrix 2

2

Enter elements of second matrix 40

30

20

10

Product of the matrices:

800 500

2000 1300

1. Write a Java program to subtract two matrices.

**import** java.util.Scanner;

**public class** Sub\_Matrix {

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

**int** i, j;

**int** mat1[][] = **new int**[3][3];

**int** mat2[][] = **new int**[3][3];

**int** mat3[][] = **new int**[3][3];

Scanner scan = **new** Scanner(System.***in***);

System.***out***.print("Enter Matrix 1 Elements : ");

**for**(i=0; i<3; i++)

{

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

{

mat1[i][j] = scan.nextInt();

}

}

System.***out***.print("Enter Matrix 2 Elements : ");

**for**(i=0; i<3; i++)

{

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

{

mat2[i][j] = scan.nextInt();

}

}

System.***out***.print("Subtracting Matrices (i.e. Matrix1 - Matrix2)...\n");

**for**(i=0; i<3; i++)

{

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

{

mat3[i][j] = mat1[i][j] - mat2[i][j];

}

}

System.***out***.print("Result of Matrix1 - Matrix2 is :\n");

**for**(i=0; i<3; i++)

{

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

{

System.***out***.print(mat3[i][j]+ " ");

}

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

}

}

}

Enter Matrix 1 Elements :

1

2

3

4

5

6

7

8

9

Enter Matrix 2 Elements : 9

8

7

6

5

4

3

2

1

Subtracting Matrices (i.e. Matrix1 - Matrix2)... Result of Matrix1 - Matrix2 is :

-8 -6 -4

-2 0 2

4 6 8

1. Write a Java program to find duplicate elements in a 1D array and find their frequency of occurrence.

**import** java.util.Scanner;

**public class** Remove\_Dupliate {

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

{

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

**int** n = arr.length;

**int**[] temp = **new int**[n];

**int** j = 0;

**for** (**int** i=0; i<n-1; i++)

**if** (arr[i] != arr[i+1]) temp[j++] = arr[i];

temp[j++] = arr[n-1];

**for** (**int** i=0; i<j; i++) arr[i] = temp[i];

**for** (**int** i=0; i<j; i++) System.***out***.print(arr[i]+" ");

}

}

1 2 3 4 5

1. Write a Java program to print every alternate number of a given array.

**import** java.util.Scanner;

**public class** Print\_Alternative {

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

{

**int** arr[] = {1,2,3,4,5,6,7,8,9,10};

**int** n = arr.length;

**for** (**int** i=0; i<n; i+=2) System.***out***.print(arr[i]+" ");

}

}

1 3 5 7 9

1. Given are two one-dimensional arrays A & B, which are sorted in ascending order. Write a Java program to merge them into single sorted array C that contains every item from arrays A & B, in ascending order.

**public class** Merge\_Two\_arrays {

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

{

**int**[] arr1 = {1, 3, 5, 7};

**int** n1 = arr1.length;

**int**[] arr2 = {2, 4, 6, 8};

**int** n2 = arr2.length;

**int**[] arr3 = **new int**[n1+n2]; **int** i = 0, j = 0, k = 0;

**while** (i<n1 && j <n2)

{

**if** (arr1[i] < arr2[j])

arr3[k++] = arr1[i++];

### else

arr3[k++] = arr2[j++];

}

**while** (i < n1)

arr3[k++] = arr1[i++];

**while** (j < n2)

arr3[k++] = arr2[j++];

System.***out***.println("Array after merging");

**for** ( i=0; i < n1+n2; i++) System.***out***.print(arr3[i] + " ");

}

}

Array after merging 1 2 3 4 5 6 7 8

1. Write a Java program to show 0-arguments constructor.

**public class** Constructor\_Zero\_Arg {

**public** Constructor\_Zero\_Arg() { System.out.println("Hello");

}

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

Constructor\_Zero\_Arg obj=**new** Constructor\_Zero\_Arg();

}

}

Hello

1. Write a Java program to show parameterized constructor.

**public class** Constructor\_Zero\_Arg {

**public** Constructor\_Zero\_Arg(String str) { System.***out***.println("Hello");

}

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

String str="Hello";

Constructor\_Zero\_Arg obj=**new** Constructor\_Zero\_Arg(str);

}

}

Hello

1. Write a Java program to show constructor overloading.

**public class** Constructor\_OverLoad {

**public** Constructor\_OverLoad() { System.***out***.println("UEMJ");

}

**public** Constructor\_OverLoad(String str) { System.***out***.println(str);

}

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

String str="UEMK";

Constructor\_OverLoad obj=**new** Constructor\_OverLoad(str);

}

} UEMK

1. Write a class, Grader, which has an instance variable, score, an appropriate constructor and appropriate methods. A method, letterGrade() that returns the letter grade as O/E/A/B/C/F.

Now write a demo class to test the Grader class by reading a score from the user, using it to create a Grader object after validating that the value is not negative and is not greater then 100. Finally, call the letterGrade() method to get and print the grade.

**import** java.util.Scanner;

**public class** Constructor\_Grade {

**public** Constructor\_Grade(**float** avg) {

**if**(avg>=90)

{

System.***out***.print("A");

}

**else if**(avg>=80 && avg<90)

{

System.***out***.print("B");

}

**else if**(avg>=60 && avg<80)

{

System.***out***.print("C");

}

**else if**(avg>=40 && avg<60)

{

System.***out***.print("D");

}

### else

{

System.***out***.print("E");

}

}

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

**float** avg;

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the %: "); avg=scanner.nextFloat(); System.***out***.print("The student Grade is: ");

Constructor\_Grade obj=**new** Constructor\_Grade(avg);

}

}

Enter the %: 91

The student Grade is: A

1. Write a class, Commission, which has an instance variable, sales; an appropriate constructor; and a method, commission() that returns the commission.

Now write a demo class to test the Commission class by reading a sale from the user, using it to create a Commission object after validating that the value is not negative. Finally, call the commission() method to get and print the commission. If the sales are negative, your demo should print the message “Invalid Input”.

**import** java.util.Scanner;

**public class** Commission {

**public** Commission(**int** avg) { System.***out***.println("Commission is "+avg);

}

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

**int** avg;

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the value: "); avg=scanner.nextInt();

Commission obj;

**if**(avg<0) System.***out***.println("Invalid Input");

### else

obj=**new** Commission(avg);

}

}

Enter the value: 999 Commission is 999

### Write a Java program to implement the concept of inheritance.

**class employee**

**{**

**private int eid; private String name; private float sal; employee()**

**{**

**this.eid = 1;**

**this.name = "Akash Chatterjee"; this.sal = 50000;**

**}**

**void print()**

**{**

**System.out.println("The id of the employee is ="+this.eid); System.out.println("The name of the employee is ="+this.name); System.out.println("The salary of the employee is ="+this.sal);**

**}**

**}**

**class programmer extends employee**

**{**

**private String develops; programmer()**

**{**

**develops = "Python"; System.out.println("Develops ="+this.develops);**

**}**

**}**

**public class Week3**

**{**

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

**{**

**programmer ob = new programmer(); ob.print();**

**}**

**}**

### Write a Java program to show method overloading.

**class box**

**{**

**private int length; private float breadth;**

**box(int length,float breadth)**

**{**

**this.length = length; this.breadth = breadth;**

**}**

**void area()**

**{ System.out.println("the area is ="+((float)this.length\*this.breadth));**

**}**

**void area(float len)**

**{ System.out.println("the area is ="+((float)len\*this.breadth)); } void area(int br)**

**{ System.out.println("the area is ="+((float)this.length\*br)); }**

**Page No**

**1**

**Object Oriented Programming using JAVA**

**Lab**

**}**

**public class Week3**

**{**

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

**{**

**box ob = new box(5,(float)9.0); ob.area(); ob.area((float)2.0); ob.area(4);**

**}**

**}**

### Write a Java program to show method overriding.

**class employee**

**{**

**private int eid; private String name; private float sal; employee()**

**{**

**this.eid = 1;**

**this.name = "Akash Chatterjee"; this.sal = 50000;**

**}**

**void print(String val)**

**{**

**System.out.println("The method wants to print="+val);**

**}**

**}**

**class programmer extends employee**

**{**

**private String develops; programmer()**

**{**

**develops = "Python";**

**}**

**void print(String val) //overridden method**

**{**

**System.out.println("Develops ="+val);**

**}**

**}**

**public class Week3**

**{**

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

**{**

**programmer ob = new programmer();**

**ob.print("Python"); //this method hides the parent method**

**}**

**}**

### Write a Java program to show method hiding.

**class employee**

**{**

**private int eid; private String name;**

**Page No**

**2**

**Object Oriented Programming using JAVA**

**Lab**

**private float sal; employee()**

**{**

**this.eid = 1;**

**this.name = "Akash Chatterjee"; this.sal = 50000;**

**}**

**void print(String val)**

**{**

**System.out.println("The method wants to print="+val);**

**}**

**}**

**class programmer extends employee**

**{**

**private String develops; programmer()**

**{**

**develops = "Python";**

**}**

**void print(String val) //overridden method**

**{**

**System.out.println("Develops ="+val);**

**}**

**}**

**public class Week3**

**{**

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

**{**

**programmer ob = new programmer();**

**ob.print("Python"); //this method hides the parent method**

**}**

**}**

### Create a general class ThreeDObject and derive the classes Box, Cube, Cylinder and Cone from it. The class ThreeDObject has methods wholeSurfaceArea ( ) and volume ( ). Override these two methods in each of the derived classes to calculate the volume and whole surface area of each type of three-dimensional objects. The dimensions of the objects are to be taken from the users and passed through the respective constructors of each derived class. Write a main method to test these classes.

**import java.util.\*; class ThreeDObject**

**{**

**private float dim1; private float dim2; private float dim3; void wholeSurfaceArea()**

**{**

**System.out.println("The surface area is = "+(float)(dim1\*dim2+dim2\*dim3+dim3\*dim1));**

**}**

**void volume()**

**{**

**System.out.println("The volume of the object = "+(float)dim1\*dim2\*dim3);**

**}**

**}**

**Page No**

**3**

**Object Oriented Programming using JAVA**

**Lab**

**class Box extends ThreeDObject{**

**Scanner sc = new Scanner (System.in); private float dim1;**

**private float dim2; private float dim3; Box(){**

**System.out.print("Enter the Dimension1 = "); dim1 = sc.nextFloat(); System.out.print("Enter the Dimension2 = "); dim2 = sc.nextFloat(); System.out.print("Enter the Dimension3 = "); dim3 = sc.nextFloat();**

**}**

**void wholeSurfaceArea(){ System.out.println("The surface area is = "+(float)(dim1\*dim2+dim2\*dim3+dim3\*dim1));**

**}**

**void volume(){**

**System.out.println("The volume of the object = "+(float)dim1\*dim2\*dim3);**

**}**

**}**

**class Cube extends ThreeDObject{**

**Scanner sc = new Scanner (System.in); private float dim1;**

**private float dim2; private float dim3; Cube()**

**{**

**System.out.print("Enter the Dimension = "); dim1 = sc.nextFloat();**

**dim2 =0;dim3=0;**

**}**

**void wholeSurfaceArea()**

**{**

**System.out.println("The surface area is = "+(float)(3\*dim1\*dim1));**

**}**

**void volume()**

**{**

**System.out.println("The volume of the object = "+(float)dim1\*dim1\*dim1);**

**}**

**}**

**class Cylinder extends ThreeDObject**

**{**

**Scanner sc = new Scanner (System.in); private float dim1;**

**private float dim2; private float dim3; Cylinder()**

**{**

**System.out.print("Enter the Radius = "); dim1 = sc.nextFloat(); System.out.print("Enter the Height = "); dim2 = sc.nextFloat();**

**dim3 = 0;**

**}**

**void wholeSurfaceArea()**

**{**

**Page No**

**4**

**Object Oriented Programming using JAVA**

**Lab**

**System.out.println("The surface area is = "+(float)(2\*3.141f\*dim1\*dim1+2\*3.141f\*dim1\*dim2));**

**}**

**void volume(){**

**System.out.println("The volume of the object = "+(float)3.141\*dim1\*dim1\*dim2);**

**}**

**}**

**class Cone extends ThreeDObject{**

**Scanner sc = new Scanner (System.in); private float dim1;**

**private float dim2; private float dim3; Cone()**

**{**

**System.out.print("Enter the Radius = "); dim1 = sc.nextFloat(); System.out.print("Enter the Height = "); dim2 = sc.nextFloat();**

**dim3 = 0;**

**}**

**void wholeSurfaceArea(){**

**System.out.println("The surface area is = "+(float)(dim1\*dim2+dim2\*dim3+dim3\*dim1));**

**}**

**void volume()**

**{**

**System.out.println("The volume of the object = "+(float)(1/3)\*3.141\*dim1\*dim1\*dim2);**

**}**

**}**

**public class Week3**

**{**

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

**{**

**Box ob = new Box(); ob.wholeSurfaceArea(); ob.volume();**

**Cylinder ob1 = new Cylinder(); ob1.wholeSurfaceArea(); ob1.volume();**

**Cube ob2 = new Cube(); ob2.wholeSurfaceArea(); ob2.volume();**

**Cone ob3 = new Cone(); ob3.wholeSurfaceArea(); ob3.volume();**

**}**

**}**

### Write a program to create a class named Vehicle having protected instance variables regnNumber, speed, color, ownerName and a method showData ( ) to show “This is a vehicle class”. Inherit the Vehicle class into subclasses named Bus and Car having individual private instance variables routeNumber in Bus and manufacturerName in Car and both of them having showData ( ) method showing all details of Bus and Car

**respectively with content of the super class’s showData ( ) method.**

**import java.util.\*;**

**Page No**

**5**

**Object Oriented Programming using JAVA**

**Lab**

**class Vehicle**

**{**

**protected String regnno; protected float speed; protected String color; protected String name; void showdata()**

**{**

**System.out.println("This is a vehicle class");**

**}**

**}**

**class Bus extends Vehicle**

**{**

**Scanner sc = new Scanner (System.in); private String routeno;**

**Bus()**

**{**

**System.out.print("Enter the routeno = "); routeno = sc.nextLine();**

**System.out.print("Enter the registration no. = "); regnno = sc.nextLine();**

**System.out.print("Enter the speed of bus = "); speed = sc.nextFloat();**

**sc.nextLine();**

**System.out.print("Enter the color of the bus = "); color = sc.nextLine();**

**System.out.print("Enter the owner name of the bus = "); name = sc.nextLine();**

**}**

**void showdata()**

**{**

**super.showdata();**

**System.out.println("The route no. of the bus = "+this.routeno); System.out.println("The registration no. of the bus = "+this.regnno); System.out.println("The Speed of the bus = "+this.speed); System.out.println("The color of the bus = "+this.color); System.out.println("The owner of the bus = "+this.name);**

**}**

**}**

**class Car extends Vehicle**

**{**

**Scanner sc = new Scanner (System.in); private String manufacturername; Car()**

**{**

**System.out.print("Enter the Manufacturer name = "); manufacturername = sc.nextLine(); System.out.print("Enter the registration no. = "); regnno = sc.nextLine();**

**System.out.print("Enter the speed of car = "); speed = sc.nextFloat();**

**sc.nextLine();**

**System.out.print("Enter the color of the car = "); color = sc.nextLine();**

**System.out.print("Enter the owner name of the car = "); name = sc.nextLine();**

**}**

**void showdata()**

**Page No**

**6**

**Object Oriented Programming using JAVA**

**Lab**

**{**

**super.showdata();**

**System.out.println("The route no. of the car = "+this.manufacturername); System.out.println("The registration no. of the car = "+this.regnno); System.out.println("The Speed of the car= "+this.speed); System.out.println("The color of the car = "+this.color); System.out.println("The owner of the car = "+this.name);**

**}**

**}**

**public class Week3**

**{**

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

**{**

**Car ob = new Car(); ob.showdata();**

**Bus obj = new Bus(); obj.showdata();**

**}**

**}**

### An educational institution maintains a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown below. Write all the classes and define the methods to create the database and retrieve individual information as and when needed.

**Write a driver program to test the classes.**

***Staff* (code, name) *Teacher* (subject, publication) is a Staff**

***Officer* (grade) is a Staff *Typist* (speed) is a Staff**

***RegularTypist* (remuneration) is a Typist *CasualTypist* (daily wages) is a Typist.**

**import java.util.\*; class Staff**

**{**

**protected String code; protected String name; void showdata()**

**{**

**System.out.println("This is employee is a staff");**

**}**

**}**

**class Teacher extends Staff**

**{**

**Scanner sc = new Scanner (System.in); private String subject;**

**private String publication; Teacher()**

**{**

**System.out.print("Enter the employee code = "); code = sc.nextLine();**

**System.out.print("Enter the employee name = "); name = sc.nextLine();**

**System.out.print("Enter the subject he/she teaches = "); subject = sc.nextLine();**

**System.out.print("Enter the publication of the teacher = "); publication = sc.nextLine();**

**}**

**Page No**

**7**

**Object Oriented Programming using JAVA**

**Lab**

**void showdata()**

**{**

**super.showdata();**

**System.out.println("The employee code of the teacher = "+this.code);**

**System.out.println("The name of the teacher = "+this.name); System.out.println("The Subject of the teacher = "+this.subject); System.out.println("The publication of the teacher =**

**"+this.publication);**

**}**

**}**

**class Officer extends Staff**

**{**

**Scanner sc = new Scanner (System.in); private String grade;**

**Officer()**

**{**

**System.out.print("Enter the employee code = "); code = sc.nextLine();**

**System.out.print("Enter the employee name = "); name = sc.nextLine();**

**System.out.print("Enter the grade of the officer = "); grade = sc.nextLine();**

**}**

**void showdata()**

**{**

**super.showdata();**

**System.out.println("The employee code of the officer = "+this.code);**

**System.out.println("The name of the officer = "+this.name); System.out.println("The Grade of the officer = "+this.grade);**

**}**

**}**

**class Typist extends Staff**

**{**

**protected float speed; void showdata()**

**{**

**System.out.println("This is employee of Typist");**

**}**

**}**

**class RegularTypist extends Typist**

**{**

**Scanner sc = new Scanner (System.in); private float remu;**

**RegularTypist()**

**{**

**System.out.print("Enter the employee code = "); code = sc.nextLine();**

**System.out.print("Enter the employee name = "); name = sc.nextLine();**

**System.out.print("Enter the typing speed of the Typist = "); speed = sc.nextFloat();**

**System.out.print("Enter the remuneration of the Typist = "); remu = sc.nextFloat();**

**}**

**void showdata()**

**{**

**Page No**

**8**

**Object Oriented Programming using JAVA**

**Lab**

**super.showdata();**

**System.out.println("The employee code of the typist = "+this.code);**

**System.out.println("The name of the typist = "+this.name); System.out.println("The typing speed of the typist =**

**"+this.speed);**

**System.out.println("The remunerisation of the typist = "+this.remu);**

**}**

**}**

**class CasualTypist extends Typist**

**{**

**Scanner sc = new Scanner (System.in); private float dailywage; CasualTypist()**

**{**

**System.out.print("Enter the employee code = "); code = sc.nextLine();**

**System.out.print("Enter the employee name = "); name = sc.nextLine();**

**System.out.print("Enter the typing speed of the Typist = "); speed = sc.nextFloat();**

**System.out.print("Enter the daily wage of the Typist = "); dailywage = sc.nextFloat();**

**}**

**void showdata()**

**{**

**super.showdata();**

**System.out.println("The employee code of the typist = "+this.code);**

**System.out.println("The name of the typist = "+this.name); System.out.println("The typing speed of the typist =**

**"+this.speed);**

**System.out.println("The daily wage of the typist = "+this.dailywage);**

**}**

**}**

**public class Week3**

**{**

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

**{**

**Teacher ob = new Teacher(); ob.showdata();**

**}**

**}**

### Create a base class Building that stores the number of floors of a building, number of rooms and it’s total footage. Create a derived class House that inherits Building and also stores the number of bedrooms and bathrooms. Demonstrate the working of the classes.

**import java.util.\*; class Building**

**{**

**protected int floors; protected int rooms; protected int footage;**

**}**

**Page No**

**9**

**Object Oriented Programming using JAVA**

**Lab**

**class House extends Building**

**{**

**private int bedrooms,bathrooms; Scanner sc = new Scanner(System.in); House()**

**{**

**System.out.println("Enetr no. of the floors ="); this.floors = sc.nextInt(); System.out.println("No. of the rooms ="); this.rooms = sc.nextInt();**

**System.out.println("No. of CCTV footage cameras ="); this.footage = sc.nextInt();**

**System.out.println("No. of the Bathrooms ="); this.bathrooms = sc.nextInt(); System.out.println("No. of the Bedrooms ="); this.bedrooms = sc.nextInt();**

**}**

**void print()**

**{**

**System.out.println("No. of the floors ="+this.floors); System.out.println("No. of the rooms ="+this.rooms); System.out.println("No. of CCTV footage cameras ="+this.footage); System.out.println("No. of the Bathrooms ="+this.bathrooms); System.out.println("No. of the Bedrooms ="+this.bedrooms);**

**}**

**}**

**public class week3**

**{**

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

**{**

**House ob = new House(); ob.print();**

**}**

**}**

### In the earlier program, create a second derived class Office that inherits Building and stores the number of telephones and tables. Now demonstrate the working of all three classes.

**import java.util.\*; class Building**

**{**

**protected int floors; protected int rooms; protected int footage; void print()**

**{**

**System.out.println("No. of the floors ="+this.floors); System.out.println("No. of the rooms ="+this.rooms); System.out.println("No. of CCTV footage cameras ="+this.footage);**

**}**

**}**

**class House extends Building**

**{**

**private int bedrooms,bathrooms;**

**Page No**

**10**

**Object Oriented Programming using JAVA**

**Lab**

**Scanner sc = new Scanner(System.in); House()**

**{**

**System.out.println("Enetr no. of the floors ="); this.floors = sc.nextInt(); System.out.println("No. of the rooms ="); this.rooms = sc.nextInt();**

**System.out.println("No. of CCTV footage cameras ="); this.footage = sc.nextInt();**

**System.out.println("No. of the Bathrooms ="); this.bathrooms = sc.nextInt(); System.out.println("No. of the Bedrooms ="); this.bedrooms = sc.nextInt();**

**}**

**void print()**

**{**

**System.out.println("No. of the floors ="+this.floors); System.out.println("No. of the rooms ="+this.rooms); System.out.println("No. of CCTV footage cameras ="+this.footage); System.out.println("No. of the Bathrooms ="+this.bathrooms); System.out.println("No. of the Bedrooms ="+this.bedrooms);**

**}**

**}**

**class Office extends Building**

**{**

**private int tables,telephones; Scanner sc = new Scanner(System.in); Office()**

**{**

**System.out.println("Enetr no. of the floors ="); this.floors = sc.nextInt(); System.out.println("No. of the rooms ="); this.rooms = sc.nextInt();**

**System.out.println("No. of CCTV footage cameras ="); this.footage = sc.nextInt(); System.out.println("No. of the Tables ="); this.tables = sc.nextInt();**

**System.out.println("No. of the Telephones ="); this.telephones = sc.nextInt();**

**}**

**void print()**

**{**

**System.out.println("No. of the floors ="+this.floors); System.out.println("No. of the rooms ="+this.rooms); System.out.println("No. of CCTV footage cameras ="+this.footage); System.out.println("No. of the Tables ="+this.tables); System.out.println("No. of the Telephones ="+this.telephones);**

**}**

**}**

**public class week3**

**{**

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

**{**

**Building ob1 = new Building(); ob1.print();**

**House ob = new House();**

**Page No**

**11**

**Object Oriented Programming using JAVA**

**Lab**

**ob.print();**

**Office ob2 = new Office(); ob2.print();**

**}**

**}**

### Write a Java program which creates a base class Num and contains an integer number along with a method shownum() which displays the number. Now create a derived class HexNum which inherits Num and overrides shownum() which displays the hexadecimal value of the number. Demonstrate the working of the classes.

**import java.util.\*; import java.lang.\*; class Num**

**{**

**protected int num; void shownum()**

**{**

**System.out.println("The number is ="+this.num);**

**}**

**}**

**class Hexnum extends Num**

**{**

**Scanner sc = new Scanner(System.in); Hexnum()**

**{**

**System.out.println("Eneter the number ="); this.num = sc.nextInt();**

**}**

**void Hexcon()**

**{**

**int n=this.num,k=0;**

**StringBuffer s = new StringBuffer(); while(n!=0)**

**{**

**if(n%16==10)**

**s.append('A'); else if (n%16==11) s.append('B'); else if (n%16==12) s.append('C'); else if (n%16==13) s.append('D'); else if (n%16==14) s.append('E'); else if (n%16==15) s.append('F'); else s.append(n%16);**

**n =(int) n/16;**

**}**

**System.out.println("The given no. in hexadecimal ="+s.reverse());**

**}**

**}**

**public class week3**

**{**

**Page No**

**12**

**Object Oriented Programming using JAVA**

**Lab**

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

**{**

**Num ob = new Num(); ob.shownum();**

**Hexnum ob1 = new Hexnum(); ob1.Hexcon();**

**}**

**}**

### Write a Java program which creates a base class Num and contains an integer number along with a method shownum() which displays the number. Now create a derived class OctNum which inherits Num and overrides shownum() which displays the octal value of the number. Demonstrate the working of the classes.

**import java.util.\*; import java.lang.\*; class Num**

**{**

**protected int num; void shownum()**

**{**

**System.out.println("The number is ="+this.num);**

**}**

**}**

**class Octnum extends Num**

**{**

**Scanner sc = new Scanner(System.in); Octnum()**

**{**

**System.out.println("Eneter the number ="); this.num = sc.nextInt();**

**}**

**void Octcon()**

**{**

**int n=this.num;**

**StringBuffer s = new StringBuffer(); while(n!=0)**

**{**

**s.append(n%8); n =(int) n/8;**

**}**

**System.out.println("The given no. in Octal ="+s.reverse());**

**}**

**}**

**public class week3**

**{**

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

**{**

**Num ob = new Num(); ob.shownum();**

**Octnum ob1 = new Octnum(); ob1.Octcon();**

**}**

**}**

**Page No**

**13**

**Object Oriented Programming using JAVA**

**Lab**

### Combine Question number 10 and 11 and have all the three classes together. Now describe the working of all classes.

**import java.util.\*; import java.lang.\*; class Num**

**{**

**protected int num; void shownum()**

**{**

**System.out.println("The number is ="+this.num);**

**}**

**}**

**class Hexnum extends Num**

**{**

**Scanner sc = new Scanner(System.in); Hexnum()**

**{**

**System.out.println("Eneter the number ="); this.num = sc.nextInt();**

**}**

**void Hexcon()**

**{**

**int n=this.num,k=0;**

**StringBuffer s = new StringBuffer(); while(n!=0)**

**{**

**if(n%16==10)**

**s.append('A'); else if (n%16==11) s.append('B'); else if (n%16==12) s.append('C'); else if (n%16==13) s.append('D'); else if (n%16==14) s.append('E'); else if (n%16==15) s.append('F'); else s.append(n%16);**

**n =(int) n/16;**

**}**

**System.out.println("The given no. in hexadecimal ="+s.reverse());**

**}**

**}**

**class Octnum extends Num**

**{**

**Scanner sc = new Scanner(System.in); Octnum()**

**{**

**System.out.println("Eneter the number ="); this.num = sc.nextInt();**

**}**

**void Octcon()**

**{**

**Page No**

**14**

**Object Oriented Programming using JAVA**

**Lab**

**int n=this.num;**

**StringBuffer s = new StringBuffer(); while(n!=0)**

**{**

**s.append(n%8); n =(int) n/8;**

**}**

**System.out.println("The given no. in Octal ="+s.reverse());**

**}**

**}**

**public class week3**

**{**

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

**{**

**Num ob = new Num(); ob.shownum();**

**Octnum ob1 = new Octnum(); ob1.Octcon();**

**Hexnum ob2 = new Hexnum(); Ob2.Hexcon();**

**}**

**}**

### Create a base class Distance which stores the distance between two locations in miles and a method travelTime(). The method prints the time taken to cover the distance when the speed is 60 miles per hour. Now in a derived class DistanceMKS, override travelTime() so that it prints the time assuming the distance is in kilometers and the speed is 100 km per second. Demonstrate the working of the classes.

**import java.util.\*; class Distance**

**{**

**protected float loc1;**

**Scanner sc = new Scanner(System.in); Distance()**

**{**

**System.out.println("Enter the distance between 2 location in Miles ="); this.loc1 = sc.nextFloat();**

**}**

**void traveltime()**

**{**

**System.out.println("The travel time is ="+(this.loc1/60)+"hr");**

**}**

**}**

**class DistanceMKS extends Distance**

**{**

**private String develops;**

**Scanner sc = new Scanner(System.in); DistanceMKS()**

**{**

**super.traveltime();**

**System.out.println("Enter the distance between 2 location in Km ="); this.loc1 = sc.nextFloat();**

**}**

**void traveltime()**

**Page No**

**15**

**Object Oriented Programming using JAVA**

**Lab**

**{**

**System.out.println("The travel time is ="+(this.loc1/100)+"sec");**

**}**

**}**

**public class week3**

**{**

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

**{**

**DistanceMKS ob1 = new DistanceMKS(); ob1.traveltime();**

**}**

**}**

### Create a base class called “vehicle” that stores number of wheels and speed. Create the following derived classes –

**“car” that inherits “vehicle” and also stores number of passengers. “truck” that inherits “vehicle” and also stores the load limit.**

**Write a main function to create objects of these two derived classes and display all**

**the information about “car” and “truck”. Also compare the speed of these two vehicles - car and truck and display which one is faster.**

**import java.util.\*; class Vehicle**

**{**

**protected int wheels; protected float speed; void showdata()**

**{**

**System.out.println("This is a vehicle class");**

**}**

**}**

**class Truck extends Vehicle**

**{**

**Scanner sc = new Scanner (System.in); private int load;**

**float getspeedtr()**

**{ return this.speed; } Truck()**

**{**

**System.out.println("Enter the no. of wheels of the truck = "); wheels = sc.nextInt();**

**System.out.println("Enter the speed of truck = "); speed = sc.nextFloat();**

**System.out.println("Enter the load limit of the truck = "); load = sc.nextInt();**

**}**

**void showdata(){**

**super.showdata();**

**System.out.print("No. of wheels of the truck = "+this.wheels); System.out.print("The speed of truck = "+this.speed); System.out.print("The load limit of the truck = "+this.load);**

**}**

**}**

**class Car extends Vehicle**

**{**

**Page No**

**16**

**Object Oriented Programming using JAVA**

**Lab**

**Scanner sc = new Scanner (System.in); private int passengers;**

**Car()**

**{**

**System.out.println("Enter the no. of wheels of the car = "); wheels = sc.nextInt();**

**System.out.println("Enter the speed of car = "); speed = sc.nextFloat();**

**System.out.println("Enter the no. of seats of the car = "); passengers = sc.nextInt();**

**}**

**float getspeedc()**

**{ return this.speed; } void showdata()**

**{**

**super.showdata();**

**System.out.println("No. of wheels of the car = "+this.wheels); System.out.println("The speed of car = "+this.speed); System.out.println("The no. of seats of the car = "+this.passengers);**

**}**

**}**

**public class week3**

**{**

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

**{**

**Car ob = new Car(); ob.showdata();**

**Truck obj = new Truck(); obj.showdata(); if(ob.getspeedc()<obj.getspeedtr())**

**System.out.println("The truck is faster than car"); else**

**System.out.println("The car is faster than the truck");**

**}**

**}**

### Write a Java program to explain “multilevel inheritance.”

**import java.util.\*;**

**class Staff //Level 0**

**{**

**protected String code; protected String name; void showdata()**

**{**

**System.out.println("This is employee is a staff");**

**}**

**}**

**class Typist extends Staff //Level 1**

**{**

**protected float speed; void showdata()**

**{**

**System.out.println("This is employee of Typist");**

**}**

**}**

**class RegularTypist extends Typist //level 2**

**Page No**

**17**

**Object Oriented Programming using JAVA**

**Lab**

**{**

**Scanner sc = new Scanner (System.in); private float remu;**

**RegularTypist()**

**{**

**System.out.print("Enter the employee code = "); code = sc.nextLine();**

**System.out.print("Enter the employee name = "); name = sc.nextLine();**

**System.out.print("Enter the typing speed of the Typist = "); speed = sc.nextFloat();**

**System.out.print("Enter the remuneration of the Typist = "); remu = sc.nextFloat();**

**}**

**void showdata()**

**{**

**super.showdata();**

**System.out.println("The employee code of the typist = "+this.code);**

**System.out.println("The name of the typist = "+this.name); System.out.println("The typing speed of the typist =**

**"+this.speed);**

**System.out.println("The remunerisation of the typist = "+this.remu);**

**}**

**}**

**public class Week3**

**{**

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

**{**

**RegularTypist ob = new RegularTypist(); ob.showdata();**

**}**

**}**

**Page No**

**18**

**Object Oriented Programming using JAVA**

**Lab**

Week 5

# Objective: Implementing the concepts of class variable, instance variable, use of “this”

**keyword, use of reference variable in Java.**

## Assignments:

1. Create a “circle” class & a “point” class. The coordinates of the circle are given and used within the “circle” class as object of the “point” class. Display the area of circle.

package Week5; import java.util.\*;

class circle{

double rad;

circle(point p1,point p2){

this.rad=Math.*sqrt*(Math.*pow*((p2.x-p1.x),2)+Math.*pow*((p2.y-p1.y),2));

}

double display\_area(){

double area=(3.14\*rad\*rad); return area;

}

}

class point{

int x,y;

point(int x, int y){

this.x=x; this.y=y;

}

}

public class circleMain{

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter the x and y axis of two point"); int a=sc.nextInt();

int b=sc.nextInt(); int c=sc.nextInt(); int d=sc.nextInt();

point p1=new point(a,b); point p2=new point(c,d);

circle mycircle=new circle(p1,p2); double disp=mycircle.display\_area();

System.*out*.println("Area of the circle is: "+disp);

}

}

Enter the x and y axis of two point 4

5

6

7

Area of the circle is: 25.120000000000005

1. Create a class called Time, which has three private instance variables – hour, min and sec. It contains a method called add( ) which takes one Time object as parameter and print the added value of the calling Time object and passes Time object. In the main method, declare two Time objects and assign values using constructor and call the add() method.

package Week5;

import java.util.Scanner; class time{

int hour,min,sec;

time(int hour,int min, int sec){ this.hour=hour; this.min=min; this.sec=sec;

}

void add\_times(time mytime){ int sec,hour,min; sec=this.sec+mytime.sec; if(sec>=60)

{

min=this.min+mytime.min+1; sec=sec-60;

}

else min=this.min+mytime.min; if(min>=60)

{

hour=this.hour+mytime.hour+1; min=min-60;

}

else hour=this.hour+mytime.hour;

System.*out*.println("\nAdded Time t1+t2 is: "+hour+" hour "+min+" min "+sec+" sec ");

}

}

public class timeMain{

public static void main(String args[]){ Scanner sc=new Scanner(System.*in*);

System.*out*.println("Enter the first Time int hh/mm/ss format: "); int h1=sc.nextInt();

int m1=sc.nextInt(); int s1=sc.nextInt();

System.*out*.println("Enter the second Time int hh/mm/ss format: "); int h2=sc.nextInt();

int m2=sc.nextInt(); int s2=sc.nextInt();

time t1=new time(h1,m1,s1); time t2=new time(h2,m2,s2);

System.*out*.println("\nTime t1 is: "+t1.hour+"hour"+t1.min+"min"+t1.sec+"sec"); System.*out*.println("Time t2 is: "+t2.hour+"hour"+t2.min+"min"+t2.sec+"sec"); t1.add\_times(t2);

}

}

Enter the first Time int hh/mm/ss format: 1

44

56

Enter the second Time int hh/mm/ss format: 1

3

45

Time t1 is: 1hour44min56sec Time t2 is: 1hour3min45sec

Added Time t1+t2 is: 2 hour 48 min 41 sec

1. Create a class called Complex, which has three private instance variables –real and imaginary. It contains a method called add( ) which takes one Complex object as parameter and print the added value of the calling Complex object and passes Complex object. In the main method, declare two Complex objects and assign values using constructor and call the add() method.

package Week5; import java. util.\*; class Complex{

int real;

int imaginary;

Complex (int real, int imaginary)

{

this.real=real;; this.imaginary=imaginary;

}

void add(Complex op2){

this.real=this.real+op2.real; this.imaginary=this.imaginary+op2.imaginary;

}

void display()

{

System.*out*.println(" The value: " + real +" + " + imaginary+"i");

}

}

public class complexDrive

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter the Real part of two numbers: "); int r1=sc.nextInt();

int r2=sc.nextInt();

System.*out*.println("Enter the Imaginary part of two numbers: "); int i1=sc.nextInt();

int i2=sc.nextInt();

Complex op=new Complex(r1,i1); Complex op1=new Complex(r2,i2);

System.*out*.println("Before add"); op.display();

op1.display(); op.add(op1);

System.*out*.println("After add"); op.display();

}

}

Enter the Real part of two numbers:

2

3

Enter the Imaginary part of two numbers:

8

6

Before add

The value: 2 + 8i The value: 3 + 6i After add

The value: 5 + 14i

1. Write a program to define a class having one 3-digit number, num as data member. Initialize and display reverse of that number.

package Week5;

import java.util.\*; class Reverse

{

int num; Reverse(int n)

{

num=n;

}

int rev(int num)

{

int rev=0; while(num>0)

{

rev=rev\*10+(num%10); num=num/10;

}

return rev;

}

}

public class RevDrive

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter a number:"); int b=sc.nextInt();

Reverse a=new Reverse(b); System.*out*.println("Reverse of "+b+" is:"); int re=a.rev(b);

System.*out*.println(re);

}

}

Enter a number:

154

Reverse of 154 is:

451

1. Write a program to define a class Student with four data members such as name, roll no., sub1, and sub2. Define appropriate methods to initialize and display the values of data members. Also calculate total marks and percentage scored by student.

package Week5;

import java.util.\*; class Student

{

int roll,sub1,sub2; String name;

int total; double per;

void Initialize(int roll,String name,int sub1,int sub2)

{

this.roll=roll; this.name=name; this.sub1=sub1; this.sub2=sub2;

}

void calculate()

{

total=sub1+sub2; per=total/2;

}

void display()

{

System.*out*.println("Name: "+ name); System.*out*.println("Roll number: "+roll); System.*out*.println("Marks of two subject: "+ sub1+" "+sub2); System.*out*.println("Total : "+total+" Percentage: "+per);

}

}

public class StudentDrive

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter your name:"); String n=sc.nextLine();

System.*out*.println("Enter your Roll number:"); int r=sc.nextInt();

System.*out*.println("Enter the marks of two subject:"); int s1=sc.nextInt();

int s2=sc.nextInt(); Student a=new Student(); a.Initialize(r,n,s1,s2); a.calculate();

a.display();

}

}

Enter your name:

Soumyadip

Enter your Roll number:

35

Enter the marks of two subject:

80

85

Name: Soumyadip Roll number: 35

Marks of two subject: 80 85

Total : 165 Percentage: 82.0

1. Write a program to define a class Employee to accept emp\_id, emp \_name, basic\_salary from the user and display the gross\_salary.

package Week5; import java.util.\*;

class Employee

{

int emp\_id;

String emp\_name; float basic\_salary;

Employee(int emp\_id, String emp\_name, float basic\_salary)

{

this.emp\_id=emp\_id; this.emp\_name=emp\_name; this.basic\_salary=basic\_salary;

}

void display()

{

float da=basic\_salary\*15/100; float hra=basic\_salary\*10/100;

float gross\_sal=basic\_salary+da+hra; System.*out*.println("YOUR DETAILS IS GIVEN BELOW: \n"); System.*out*.println ("Employee Id= "+emp\_id); System.*out*.println ("Emplyee Name= "+emp\_name); System.*out*.println ("Gross Salary= "+gross\_sal);

}

}

public class EmpDrive

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println ("Enter Employee id"); int id = sc.nextInt();

System.*out*.println ("Enter Employee Name"); String name = sc.nextLine();

name = sc.nextLine();

System.*out*.println ("Enter Basic Salary"); float sal = sc.nextFloat();

Employee e = new Employee(id, name, sal); e.display();

}

}

Enter Employee id 1

Enter Employee Name Soumyadip

Enter Basic Salary 35000

YOUR DETAILS IS GIVEN BELOW:

Employee Id= 1

Emplyee Name= Soumyadip Gross Salary= 43750.0

1. Write a program to define a class Fraction having data members numerator and denominator. Initialize three objects using different constructors and display its fractional value.

package Week5; import java.util.\*;

class Fraction

{

double numerator,denominator; Fraction (int a, double b)

{

numerator=a; denominator=b;

}

Fraction (int x, int y)

{

numerator=x; denominator=y;

}

Fraction(double m, double n)

{

numerator=m; denominator=n;

}

void display()

{

double fraction=numerator/denominator; System.*out*.println ("Fraction = "+fraction);

}

}

public class FractionDrive

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter the Numerator of three numbers:");

int n1=sc.nextInt(); int n2=sc.nextInt();

double n3=sc.nextDouble();

System.*out*.println("Enter the Denominator of three numbers:"); double d1=sc.nextDouble();

int d2=sc.nextInt();

double d3=sc.nextDouble(); Fraction f1 = new Fraction(n1,d1); f1.display();

Fraction f2 = new Fraction(n2,d2); f2.display();

Fraction f3 = new Fraction(n3,d3); f3.display();

}

}

Enter the Numerator of three numbers:

55

77

91

Enter the Denominator of three numbers:

3

5

6

Fraction = 18.333333333333332

Fraction = 15.4

Fraction = 15.166666666666666

1. Write a program to define a class Item containing code and price. Accept this data for five objects using array of objects. Display code, price in tabular form and also, display total price of all items.

package Week5;

import java.util.\*; class Item

{

int price; int code;

Item(int m,int n)

{

code=m; price=n;

}

void display()

{

System.*out*.print(code+" "+price); System.*out*.println();

}

}

public class ItemDrive

{

public static void main(String args[])

{

Scanner sc= new Scanner(System.*in*); int a,c,sum=0;

Item[] obj=new Item[5]; for(int i=0;i<5;i++)

{

System.*out*.println("Enter Code :"); c=sc.nextInt(); System.*out*.println("Enter Price :"); a=sc.nextInt();

obj[i]=new Item(c,a);

}

for(int i=0;i<5;i++)

{

sum=sum+obj[i].price;

}

System.*out*.println("Code Price "); for(int i=0;i<5;i++)

{

obj[i].display();

}

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

}

}

Enter Code : 101

Enter Price : 1000

Enter Code : 201

Enter Price : 300

Enter Code : 301

Enter Price : 9000

Enter Code : 401

Enter Price : 4400

Enter Code : 501

Enter Price : 9000

|  |  |
| --- | --- |
| Code | Price |
| 101 | 1000 |
| 201 | 300 |
| 301 | 9000 |
| 401 | 4400 |
| 501 | 9000 |

Total Cost: 23700

1. Write a program to define a class Tender containing data members cost and company name. Accept data for five objects and display company name for which cost is minimum. package Week5;

import java.util.\*; class Tender

{

int cost; String name;

Tender(String a,int b)

{

name=a; cost=b;

}

void display()

{

System.*out*.println(name+"\t"+cost);

}

}

public class MinDrive

{

public static void main(String args[])

{

int cost,k=-1; String name;

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter the number of tenders"); int n=sc.nextInt();

Tender obj[]=new Tender[n];

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

{

System.*out*.print("Enter the Name of Company:"); name=sc.nextLine();

name=sc.nextLine(); System.*out*.print("Enter the Cost:"); cost=sc.nextInt();

obj[i]=new Tender(name,cost);

}

System.*out*.println("Company Name Cost"); for(int i=0;i<n;i++)

{

obj[i].display();

}

int min=obj[0].cost; for(int i=1;i<n;i++)

{

if(obj[i].cost<min)

{

k=i; min=obj[i].cost;

}

}

System.*out*.println("Minimum = "+min);

}

}

Enter the number of tenders 2

Enter the Name of Company:TC Enter the Cost:90000

Enter the Name of Company:IC Enter the Cost:100000 Company Name Cost

TC 90000

IC 100000

Minimum = 90000

1. Write a program to define a class 'employee' with data members as empid, name and salary. Accept data for 5 objects using Array of objects and print it.

package Week5;

import java.util.\*; class Item1

{

int p,eid; String c;

Item1(String m,int n,int e)

{

c=m; p=n; eid=e;

}

void display()

{

System.*out*.print(eid+" "+c + " " + p); System.*out*.println();

}

}

public class SalDrive

{

public static void main(String args[])

{

Scanner sc= new Scanner(System.*in*); int b,c;

String a;

Item1[] obj=new Item1[5]; for(int i=0;i<5;i++)

{

System.*out*.print("Enter ID :"); c=sc.nextInt(); System.*out*.print("Enter name :"); a=sc.nextLine();

a=sc.nextLine(); System.*out*.print("Enter salary :"); b=sc.nextInt();

obj[i]=new Item1(a,b,c);

}

System.*out*.println("Employee ID Name Salary "); for(int i=0;i<5;i++)

{

obj[i].display();

}

}

}

Enter ID :1

Enter name :Soumyadip Enter salary :35000

Enter ID :2

Enter name :Sayan Enter salary :30000

Enter ID :3

Enter name :Arko Enter salary :33000

Enter ID :4

Enter name :Sapta Enter salary :34000

Enter ID :5 Enter name :Avi

Enter salary :35000 Employee ID Name Salary

1 Soumyadip 35000 2 Sayan 30000

3 Arko 33000

4 Sapta 34000

5 Avi 35000

1. Define a class called circle that contains:

* Two private instance variables: radius (of type double) and color (of type String),
* Initialize the variables radius and color with default value of 1.0 and "red", respectively

using default constructor.

* Include a second constructor that will use the default value for color and sets the radius to the value passed as parameter.
* Two public methods: getRadius() and getArea() for returning the radius and area of

the circle

* Invoke the above methods and constructors in the main.

package Week5;

import java.util.\*; class AB

{

private double radius; private String color; AB()

{

radius=1.0;

color="red";

}

AB(double a,String col)

{

radius=a; color=col;

}

double getRadius()

{

return radius;

}

double getArea()

{

double area=3.14\*radius\*radius; return area;

}

}

public class CDrive

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter the Radius:"); double rad=sc.nextDouble(); System.*out*.println("Enter the color:"); String clr=sc.nextLine(); clr=sc.nextLine();

AB a=new AB();

AB b=new AB(rad,clr); double q=a.getRadius();

System.*out*.println("Value of radius when we call getRadius() with non parameterized constructor = "+q);

double g=b.getRadius();

System.*out*.println("Value of radius when we call getRadius() with parameterized constructor = "+g);

double ar=b.getArea(); System.*out*.println("Area = "+ar); System.*out*.println("Colour = "+clr);

}

}

Enter the Radius:

4

Enter the color:

Red

Value of radius when we call getRadius() with non parameterized constructor = 1.0

Value of radius when we call getRadius() with parameterized constructor = 4.0 Area = 50.24

Colour = Red

1. Write a program which will accept an integer from the user and pass the value to a method called PrintNumberInWord that will print "ONE", "TWO",... , "NINE", "ZERO" if the integer variable "number" is 1, 2,... , 9, or 0, respectively.

package Week5;

import java.util.Scanner; class number

{

public static void numberToWord(int num, String val) {

String ones[] = {" ", " ONE", " TWO", " THREE", " FOUR", " FIVE", " SIX", " SEVEN", " EIGHT", " NINE",

" TEN", " ELEVEN", " TWELVE", " THIRTEEN", " FOURTEEN", " FIFTEEN", " SIXTEEN", " SEVENTEEN", " EIGHTEEN", " NINETEEN"

};

String tens[] = {" ", " ", " TWENTY", " THIRTY", " FOURTY", " FIFTY", " SIXTY", " SEVENTY", " EIGHTY", " NINETY"};

if (num > 19) {

System.*out*.print(tens[num / 10] + " " + ones[num % 10]);

} else {

System.*out*.print(ones[num]);

}

if (num > 0) { System.*out*.print(val);

}

}

}

public class NumDrive

{

public static void main(String[] args) { int number = 0;

Scanner scanner = new Scanner(System.*in*); number n=new number();

System.*out*.print("Please type a number between 0 and 999 OR type -1 to exit: "); number = scanner.nextInt();

while(number!=-1){

if(number>=0 && number<=999){ if(number==0){

System.*out*.print("NUMBER AFTER CONVERSION:\tZERO");

} else {

System.*out*.print("NUMBER AFTER CONVERSION:\t"); n.*numberToWord*(((number / 100) % 10), " HUNDRED");

n.*numberToWord*((number % 100), " ");

}

} else{

System.*out*.print("NUMBER OUT OF RANGE");

}

System.*out*.print("\nPlease type a number between 0 and 999 OR type -1 to exit: "); number = scanner.nextInt();

}

}

}

Please type a number between 0 and 999 OR type -1 to exit: 191 NUMBER AFTER CONVERSION: ONE HUNDRED NINETY ONE

1. Design a class named Account that contains:
2. A private int data field named id for the account (default 0).
3. A private double data field named balance for the account (default 0).
4. A private double data field named annualInterestRate that stores the cur-rent interest rate (default 0). Assume all accounts have the same interest rate.
5. A private Date data field named dateCreated that stores the date when the account was created.
6. A no-arg constructor that creates a default account.
7. A constructor that creates an account with the specified id and initial balance.
8. The accessor and mutator methods for id,balance, and annualInterestRate.
9. The accessor method for dateCreated.
10. A method named getMonthlyInterestRate() that returns the monthly interest rate.
11. A method named getMonthlyInterest() that returns the monthly interest.
12. A method named withdraw that withdraws a specified amount from the account.
13. A method named deposit that deposits a specified amount to the account.

package Week5;

class Account { private int id = 0;

private double balance = 0.0;

private static double *annualInterestRate* = 0.0; private java.util.Date dateCreated;

public Account() {

dateCreated = new java.util.Date();

}

public Account(int id, double balace) { this();

this.id = id; this.balance = balance;

}

public int getId() { return this.id;

}

public double getBalance() { return this.balance;

}

public double getAnnualInterestRate() { return *annualInterestRate*;

}

public String getDateCreated() { return this.dateCreated.toString();

}

public void setId(int id) { this.id = id;

}

public void setBalance(double balance) { this.balance = balance;

}

public void setAnnualInterestRate(double annualInterestRate) { this.*annualInterestRate* = annualInterestRate;

}

public double getMonthlyInterestRate() { return (*annualInterestRate* / 100) / 12 ;

}

public double getMonthlyInterest() {

return balance \* getMonthlyInterestRate();

}

public void withdraw(double amount) { this.balance -= amount;

}

public void deposit(double amount) { this.balance += amount;

}

}

public class AccDrive {

public static void main(String[] args) {

Account account = new Account(1122, 20000); account.setAnnualInterestRate(4.5); account.withdraw(2500.0); account.deposit(3000.0);

System.*out*.println("Balance: $" + account.getBalance()); System.*out*.println("Monthly Interest: " + account.getMonthlyInterest()); System.*out*.println("Date Created: " + account.getDateCreated());

}

}

Balance: $500.0 Monthly Interest: 1.875

Date Created: Sun Aug 25 10:29:13 IST 2019

1. Write a test program that prompts the user to enter the investment amount (e.g., 1000) and the interest rate (e.g., 9%), and print a table that displays future value for the years from 1 to 30, as shown below:

The amount invested: 1000 Annual interest rate: 9% Years Future Value

1 1093.8

2 1196.41

...

29 13467.25

30 14730.57

package Week5; import java.util.\*; class Amt

{

Double x;

Double sum=1000.0; Amt(Double x)

{

this.x=x;

}

void interest()

{

System.*out*.println("Years future\_value");

for(int i=1;i<=30;i++)

{

System.*out*.println(i+". "+sum(x,0.09/12,i));

}

}

Double sum(Double tot,double rate,int years){ return x\*Math.*pow*(1+rate,years\*12);

}

}

class ADrive

{

public static void main(String[] args) { Amt ob=new Amt(100.0); ob.interest();

}

}

Years future\_value

1........109.38068976709839

2........119.64135293926222

3........130.86453709165366

4........143.1405333313711

5........156.56810269415706

6........171.25527068212796

7........187.32019633462298

8........204.89212282389357

9........224.1124172232252

10........245.13570781248114

11........268.1311280707507

12........293.28367736408916

13........320.7957092751521

14........350.888559548417

15........383.80432674789427

16........419.80781995281484

17........459.1886891606074

18........502.2637555363697

19........549.379560255814

20........600.9151524472612

21........657.2851386618252

22........718.9430184049334

23........786.3848325637133

24........860.1531540820313

25........940.8414529883785

26........1029.098870893479

27........1125.6354433687086

28........1231.2278122196296

29........1346.7254736101859

30........1473.057612304044

1. Write method headers for the following methods:
2. Computing a sales commission, given the sales amount and the commission rate.
3. Printing the calendar for a month, given the month and year.
4. Computing a square root.
5. Testing whether a number is even, and returning true if it is.
6. Printing a message a specified number of times.
7. Computing the monthly payment, given the loan amount, number of years, and annual interest rate.
8. public static double getCommission(double salesAmount, double commissionRate)
9. public static void printCalendar(int month, int year)
10. public static double sqrt(double value)
11. public static boolean isEven(int value)
12. public static void printMessage(String message, int times)
13. public static double monthlyPayment(double loan, int numberOfYears, double annualInterestRate)
14. Write a program that reads ten numbers, computes their average, and finds out how many numbers are above the average. [Use this keyword]

package Week5; import java.util.\*;

class B

{

static int *a*[],*n*; B(int a[],int n)

{

this.*a*=a; this.*n*=n;

}

void calc()

{

int avg=0; int c=0;

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

avg=avg+*a*[i];

avg=avg/*n*; System.*out*.println("Average is :"+avg); for(int i=0;i<*n*;i++)

{

if(*a*[i]>avg) c++;

}

if(c>0)

else

}

}

System.*out*.println("There are "+c+" numbers that are above the average "); System.*out*.println("There are no numbers that are below the average ");

class AvgDrive

{

public static void main(String[] args) { Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter n :");

int n=sc.nextInt(); int a[]=new int[n]; for(int i=0;i<n;i++)

{

System.*out*.println("Enter number :"); a[i]=sc.nextInt();

}

B obj=new B(a,n); obj.calc();

}

}

Enter n :

3

Enter number : 55

Enter number : 66

Enter number :

77

Average is :66

There are 1 numbers that are above the average

1. Write a program that reads ten integers and displays them in the reverse of the order in which they were read.

package Week5; class Num

{

void rev(int[] num)

{

for(int i=9;i>=0;i--)

{

System.*out*.println ("in reverse order"); System.*out*.println (num[i]);

}

}

}

public class ReverseDrive

{

public static void main(String[] args)

{

Num ob=new Num();

int[] num={0,1,2,3,4,5,6,7,8,9};

ob.rev(num);

}

}

in reverse order 9

in reverse order 8

in reverse order 7

in reverse order 6

in reverse order 5

in reverse order 4

in reverse order 3

in reverse order 2

in reverse order 1

in reverse order 0

1. Write a program to demonstrate use of 'this' keyword.

package Week5;

class Student1{

int rollno; String name; float fee;

Student1(int rollno,String name,float fee){ this.rollno=rollno; this.name=name;

this.fee=fee;

}

void display()

{

System.out.println(rollno+" "+name+" "+fee);

}

}

public class ThisDrive{

public static void main(String args[]){

Student1 s1=new Student1(100,"Soumya",3000f); Student1 s2=new Student1(101,"Avi",2000f); s1.display();

s2.display();

}

}

100 Soumya 3000.0

101 Avi 2000.0

1. Write a program to demonstrate use of 'static' keyword.

package Week5; class Demo

{

static void m1()

{

System.*out*.println("Demo of static");

}

}

public class StaticDrive

{

public static void main(String[] args)

{

Demo.*m1*();

}

}

Demo of static

1. Write a program to accept value of apple sales for each day of the week (using array of type float) and then, calculate the average sale of the week.

package Week5;

class Sales

{

int x;

Double sum=0.0,avg;

void week(double[] sales)

{

for(int i=1;i<=7;i++) { sum=sum+sales[i-1];

}

System.*out*.println ("Sum = "+sum); avg=sum/7.0;

System.*out*.println ("Average sale of week="+avg);

}

}

public class SalesDrive

{

public static void main(String[] args) { Sales obj=new Sales();

double[] sales={189.5,204.3,123.8,223.7,399.6,566.3,992.2};

obj.week(sales);

}

}

Sum = 2699.4

Average sale of week=385.62857142857143

1. Write program, which finds the sum of numbers formed by consecutive digits. Input : 2415 output : 24+41+15=80.

package Week5;

import java.util.\*; class Digit

{

int x;

int y=0,z=0,sum=0,m=0; Digit(int x)

{

this.x=x;

}

void num()

{

while (x>9)

{

y=x%10;

x=x/10; z=x%10;

m=z\*10; sum=sum+y+m;

}

System.*out*.println("sum of numbers formed by consecutiv digits="+sum);

}

}

public class DigitDrive

{

public static void main(String[] args) { Scanner sc=new Scanner(System.*in*); System.*out*.print("Enter a number:"); int i=sc.nextInt();

Digit obj=new Digit(i); obj.num();

}

}

Enter a number:2415

sum of numbers formed by consecutiv digits=80

**JAVA-Week-6**

**Objective: To revise inheritance and to understand the concepts of Abstract class & Interface in Java.**

### Assignments:

1. **Design an abstract class having two methods. Create Rectangle and Triangle classes by inheriting the shape class and override the above methods to suitably implement for Rectangle and Triangle class.**

**abstract class** Area\_Peri{

**public abstract int** area(**int** w,**int** l);

**public abstract double** perimeter(**int** a,**int** b);

}

**class** Rectangle **extends** Area\_Peri{

**public int** area(**int** w, **int** l) {

**return** w\*l;

}

**public double** perimeter(**int** w, **int** l) {

**return** 2\*(w+l);

}

}

**class** Triangle **extends** Area\_Peri{

//Isosceles Triangle

**public int** area(**int** h,**int** b) {

**return** (h\*b)/2;

}

**public double** perimeter(**int** h, **int** b) {

**return** (h+h+b);

}

}

**public class** Abstract\_Rect\_Tri {

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

Area\_Peri obj=**new** Rectangle();

System.***out***.println("Area of the Rectangle "+obj.area(10,20)); System.***out***.println("perimeter of the Rectangle "+obj.perimeter(10,20));

Area\_Peri obj1=**new** Triangle();

System.***out***.println("Area of the Triangle "+obj1.area(10,20)); System.***out***.println("perimeter of the Triangle "+obj1.perimeter(20,10));

}

}

Area of the Rectangle200 perimeter of the Rectangle60.0 Area of the Triangle100 perimeter of the Triangle50.0

### Write a program in Java to illustrate the use of interface in Java.

**interface** Example{

**public void** show();

}

**class** Test\_Example **implements** Example{

**public void** show() {

System.***out***.print("This is Interface");

}

}

**public class** Interface\_Test {

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

}

}

This is Interface

Example obj=**new** Test\_Example(); obj.show();

### Create a general class ThreeDObject and derive the classes Box, Cube, Cylinder and Cone from it. The class ThreeDObject has methods wholeSurfaceArea ( ) and volume( ). Override these two methods in each of the derived classes to calculate the volume and whole surface area of each type of three- dimensional objects. The dimensions of

**the objects are to be taken from the users and passed through the respective constructors of each derived class. Write a main method to test these classes.**

**import** java.util.Scanner;

**abstract class** ThreeDObject{

**abstract int** SurfaceArea(**int** h, **int** w, **int** l);

**abstract int** volume(**int** h, **int** w, **int** l);

}

**class** Box **extends** ThreeDObject{

**int** SurfaceArea(**int** h, **int** w, **int** l) {

**return** (2\*(h\*w)+2\*(h\*l)+2\*(w\*l));

}

**int** volume(**int** h, **int** w, **int** l) {

**return** (h\*w\*l);

}

}

**class** Cube **extends** ThreeDObject{

**int** SurfaceArea(**int** a,**int** b,**int** c) {

**return** (6\*a\*b);

}

**int** volume(**int** a,**int** b,**int** c) {

**return** (a\*b\*c);

}

}

**class** Cone **extends** ThreeDObject{

**int** SurfaceArea(**int** h,**int** r,**int** a) {

**return** (**int**) (Math.***PI***\*a\*(r+Math.*sqrt*(Math.*pow*(h, 2)+Math.*pow*(r, 2))));

}

**int** volume(**int** h,**int** r,**int** a) {

**return** (**int**) (Math.***PI***\*a\*r\*(h/3));

}

}

**class** Cylinder **extends** ThreeDObject{

**int** SurfaceArea(**int** h,**int** r,**int** a) {

**return** (**int**) (2\*Math.***PI***\*r\*h+2\*Math.***PI***\*r\*a);

}

**int** volume(**int** h,**int** r,**int** a) {

**return** (**int**) (Math.***PI***\*r\*a\*h);

}

}

**public class** ThreeD\_Class {

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

{

ThreeDObject obj=**new** Box(); ThreeDObject obj1=**new** Cube(); ThreeDObject obj2=**new** Cone(); ThreeDObject obj3=**new** Cylinder();

System.***out***.println("Box surface area "+obj.SurfaceArea(10,20,30)); System.***out***.println("Cube surface area "+obj1.SurfaceArea(10,10,10)); System.***out***.println("Cone surface area "+obj2.SurfaceArea(10,20,20)); System.***out***.println("Cylinder surface area "+obj3.SurfaceArea(10,20,20));

System.***out***.println("Box volume " +obj.volume(10,20,30)); System.***out***.println("Cube volume "+obj1.volume(10,10,10)); System.***out***.println("Cone volume "+obj2.volume(10,20,20));

System.***out***.println("Cylinder volume "+obj3.volume(10,20,20));

}

}

Box surface area 2200 Cube surface area 600 Cone surface area 2661 Cylinder surface area 3769 Box volume 6000

Cube volume 1000

Cone volume 3769

Cylinder volume 12566

### Write a program to create a class named Vehicle having protected instance variables regnNumber, speed, color, ownerName and a method showData ( ) to show “This is a vehicle class”. Inherit the Vehicle class into subclasses named Bus and Car having individual private instance variables routeNumber in Bus and manufacturerName in Car and both of them having showData ( ) method showing all details of Bus and Car respectively with content of the super class’s showData ( ) method.

**class** Vehicle\_Main {

**protected int** regnNumber; **protected int** speed; **protected** String color; **protected** String ownerName;

**public** Vehicle\_Main(**int** regnNumber, **int** speed, String color, String ownerName) {

**this**.regnNumber = regnNumber;

**this**.speed = speed;

**this**.color = color;

**this**.ownerName = ownerName;

}

}

**class** Bus **extends** Vehicle\_Main{

**int** routeNumber;

**public** Bus(**int** regnNumber, **int** speed, String color, String ownerName, **int** routeNumber) {

**super**(regnNumber, speed, color, ownerName);

**this**.routeNumber = routeNumber;

}

**public** String ShowData() {

**return** "Bus [routeNumber=" + routeNumber + ", regnNumber=" + regnNumber + ", speed=" + speed + ", color="

+ color + ", ownerName=" + ownerName + "]";

}

}

**class** Car **extends** Vehicle\_Main{ String manufacturerName ;

**public** Car(**int** regnNumber, **int** speed, String color, String ownerName, String manufacturerName ) {

**super**(regnNumber, speed, color, ownerName);

**this**.manufacturerName = manufacturerName ;

}

**public** String ShowData() {

**return** "Car [manufacturerName=" + manufacturerName + ", regnNumber=" + regnNumber + ", speed=" + speed + ", color="

+ color + ", ownerName=" + ownerName + "]";

}

}

**public class** Vehicle\_Drive {

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

Bus obj=**new** Bus(1010,70,"Black","Soumyadip",199); System.***out***.println(obj.ShowData());

Car obj1=**new** Car(1010,70,"Black","Soumyadip","TATA"); System.***out***.println(obj1.ShowData());

}

}

Bus [routeNumber=199, regnNumber=1010, speed=70, color=Black, ownerName=Soumyadip]

Car [manufacturerName=TATA, regnNumber=1010, speed=70, color=Black, ownerName=Soumyadip]

### Create three interfaces, each with two methods. Inherit a new interface from the three, adding a new method. Create a class by implementing the new interface and also inheriting from a concrete class. Now write four methods, each of which takes one of the four interfaces as an argument. In main ( ), create an object of your class and pass it to each of the methods.

**interface** Test1{

**public void** show1();

**public void** show2();

}

**interface** Test2{

**public void** show3();

**public void** show4();

}

**interface** Test3{

**public void** show5();

**public void** show6();

}

**interface** Test4{

**public void** show7();

**public void** show8();

}

**interface** MultiTest **extends** Test1,Test2,Test3,Test4{

**public void** newMethod();

}

**class** Multi **implements** MultiTest{

**public void** show1() {

System.***out***.println("Mehtod 1");

}

**public void** show2() {

System.***out***.println("Mehtod 2");

}

**public void** show3() {

System.***out***.println("Mehtod 3");

}

**public void** show4() {

System.***out***.println("Mehtod 4");

}

**public void** show5() {

System.***out***.println("Mehtod 5");

}

**public void** show6() {

System.***out***.println("Mehtod 6");

}

**public void** show7() {

System.***out***.println("Mehtod 7");

}

**public void** show8() {

System.***out***.println("Mehtod 8");

}

**public void** newMethod() { System.***out***.println("Mehtod newMethod()");

}

}

**public class** Multi\_Interface {

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

MultiTest obj=**new** Multi(); obj.show1();

obj.show2();

obj.show3();

obj.show4();

obj.show5();

obj.show6();

obj.show7();

obj.show8(); obj.newMethod();

}

}

Mehtod 1

Mehtod 2

Mehtod 3

Mehtod 4

Mehtod 5

Mehtod 6

Mehtod 7

Mehtod 8

Mehtod newMethod()

### Create an interface Department containing attributes deptName and deptHead. It also has abstract methods for printing the attributes. Create a class hostel containing hostelName, hostelLocation and numberofRooms. The class contains methods for getting and printing the attributes. Then write Student class extending the Hostel class and implementing the Department interface. This class contains attributes studentName, regdNo, electiveSubject and avgMarks. Write suitable getData and printData methods for this class. Also implement the abstract methods of the Department interface. Write a driver class to test the Student class. The program should be menu driven containing the options:

1. **Admit new student**
2. **Migrate a student**
3. **Display details of a student**

**For the third option a search is to be made on the basis of the entered registration number.**

**import** java.util.Scanner;

**interface** Department{

**public final** String ***deptName***="CSE";

**public final** String ***deptHead***="XYZ\_Sir";

**public abstract** String printData();

}

**class** Hostel{

**protected** String hostelName,hostelLocation;

**protected int** numberofRooms;

**public** Hostel(String hostelName, String hostelLocation, **int** numberofRooms) {

**this**.hostelName = hostelName; **this**.hostelLocation = hostelLocation; **this**.numberofRooms = numberofRooms;

}

}

**class** Student **extends** Hostel **implements** Department{

**protected** String studentName, regdNo, electiveSubject;

**protected int** avgMarks;

**public** Student(String hostelName, String hostelLocation, **int** numberofRooms, String studentName, String regdNo,

String electiveSubject, **int** avgMarks) { **super**(hostelName, hostelLocation, numberofRooms); **this**.studentName = studentName;

**this**.regdNo = regdNo; **this**.electiveSubject = electiveSubject; **this**.avgMarks = avgMarks;

}

**public** String printData() {

**return** "Student [studentName=" + studentName + ", regdNo=" + regdNo + ", electiveSubject=" + electiveSubject

+ ", avgMarks=" + avgMarks + ", hostelName=" + hostelName + ", hostelLocation=" + hostelLocation

+ ", numberofRooms=" + numberofRooms + ", deptName=" +

***deptName*** + ", deptHead=" + ***deptHead*** + "]";

}

}

**public class** Student\_Interface {

**public static void** main(String args[]) { Scanner sc=**new** Scanner(System.***in***);

String studentName, regdNo, electiveSubject;

**int** avgMarks,numberofRooms; String hostelName,hostelLocation; Student[] obj=**new** Student[10]; **int** x=1,i=1;

**while**(x!=0)

{

System.***out***.println("1: Admit"); System.***out***.println("2: Migrate"); System.***out***.println("3: Show"); **int** z=sc.nextInt();

**if**(z==1)

{

System.***out***.println("Enter Student Name"); studentName=sc.next();

regdNo="UEMK3RD"+i; System.***out***.println("Enter elected Subject"); electiveSubject=sc.next(); System.***out***.println("Enter avgMarks"); avgMarks=sc.nextInt(); System.***out***.println("Enter hostelName"); hostelName=sc.next(); System.***out***.println("Enter hostelLocation"); hostelLocation=sc.next(); System.***out***.println("Enter numberofRooms"); numberofRooms=sc.nextInt();

obj[i++]=**new** Student( hostelName, hostelLocation, numberofRooms, studentName, regdNo,electiveSubject, avgMarks);

System.***out***.println("Student Added::Reg Id"+regdNo);

}

**else if**(z==2)

{

System.***out***.print("Enter the last digit of the id ");

**int** o=sc.nextInt(); obj[o]=**null**;

}

**else if**(z==3){

System.***out***.print("Enter the last digit of the id "); **int** o=sc.nextInt(); System.***out***.println(obj[o].printData());

}

}

1: Admit

2: Migrate

3: Show

1

Enter Student Name Soumyadip

Enter elected Subject Java

Enter avgMarks 80

Enter hostelName UEMK\_HOSTEL

Enter hostelLocation

}

### else

{

}

}

x=0;

Newtown

Enter numberofRooms 3

Student Added::Reg IdUEMK3RD1 1: Admit

2: Migrate

3: Show

3

Enter the last digit of the id 1

Student [studentName=Soumyadip, regdNo=UEMK3RD1, electiveSubject=Java, avgMarks=80, hostelName=UEMK\_HOSTEL, hostelLocation=Newtown, numberofRooms=3, deptName=CSE, deptHead=XYZ\_Sir]

### Create an interface called Player. The interface has an abstract method called play() that displays a message describing the meaning of “play” to the class. Create classes called Child, Musician, and Actor that all implement Player. Create an application that demonstrates the use of the classes (UsePlayer.java)

**interface** Player{

**abstract public void** play();

}

**class** Child **implements** Player{

**public void** play() {

System.***out***.println("This is Child");

}

}

**class** Musician **implements** Player{

**public void** play() {

System.***out***.println("This is Musician");

}

}

**class** Actor **implements** Player{

**public void** play() {

System.***out***.println("This is Actor ");

}

}

**public class** UsePlayer {

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

Player obj=**new** Child(); Player obj1=**new** Musician(); Player obj2=**new** Actor(); obj.play();

obj1.play();

obj2.play();

}

}

This is Child This is Musician This is Actor

### Create an abstract class Accounts with the following details: Data Members:

**(a) Balance (b) accountNumber (c) accountHoldersName (d) address Methods:**

1. **withdrawl()- abstract**
2. **deposit()- abstract**
3. **display() to show the balance of the account number**

**Create a subclass of this class SavingsAccount and add the following details: Data Members:**

**(a) rateOfInterest Methods:**

**(a) calculateAount()**

**abstract class** Account{

**protected** String dipositor; **protected** String address; **protected long** ac; **protected int** ammount;

**public** Account(String dipositor, String address, **long** ac, **int** ammount) {

**this**.dipositor = dipositor;

**this**.address = address;

**this**.ac = ac;

**this**.ammount = ammount;

}

**abstract public void** withdraw(**int** am); **abstract public void** diposit(**int** a); **abstract public** String display();

}

**class** SavingsAccount **extends** Account{

**private int** roi;

**public** SavingsAccount (String dipositor, String address, **long** ac, **int** ammount) {

**super**(dipositor, address, ac, ammount);

**this**.roi = 5;

}

**public void** withdraw(**int** am)

{

**if**(am>ammount)

{

System.***out***.println("Invalid ammount");

}

### else{

ammount-=am;

System.***out***.println("Ammount debited "+ammount); System.***out***.println("Updated balance "+ammount);

}

}

**public void** diposit(**int** a)

{

ammount+=a;

System.***out***.println("Ammount debited "+ammount); System.***out***.println("Updated balance "+ammount);

}

**public void** calculateAmount()

{

**int** sum=ammount\*(1+roi\*12); System.***out***.println("Intrestfor 12 months :"+sum); System.***out***.println("Recent balance "+ammount);

}

**public** String display() {

**return** "SavingsAccount [Intrest Rate=" + roi + "%, Dipositor=" + dipositor + ", Address="

+ address + ", Ac/No.=" + ac

+ ", Ammount=" + ammount + "]";

}

}

**public class** Abstract\_Bank {

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

SavingsAccount obj=**new** SavingsAccount ("Soumyadip", "Birati",198982,2000); obj.diposit(500);

obj.withdraw(300); obj.calculateAmount(); System.***out***.print(obj.display());

}

}

Ammount debited 2500

Updated balance 2500

Ammount debited 2200

Updated balance 2200

Intrestfor 12 months :134200

Recent balance 2200

SavingsAccount [Intrest Rate=5%, Dipositor=Soumyadip, Address=Birati, Ac/No.=198982, Ammount=2200]

### Create an abstract class MotorVehicle with the following details: Data Members:

**(a) modelName (b)modelNumber (c) modelPrice Methods:**

1. **display() to show all the details**

**Create a subclass of this class Carthat inherits the class MotorVehicle and add the following details: Data Members:**

1. **discountRate Methods:**
2. **display() method to display the Car name, model number, price and the discount rate.**
3. **discount() method to compute the discount.**

**abstract class** MotorVehicle {

**protected** String modelName; **protected int** mmdelPrice; **protected int** modelNumber;

**public** MotorVehicle(String modelName, **int** mmdelPrice, **int** modelNumber) {

**super**();

**this**.modelName = modelName; **this**.mmdelPrice = mmdelPrice; **this**.modelNumber = modelNumber;

}

**abstract public** String display();

}

**class** Carthat **extends** MotorVehicle{

**double** discountRate;

**public** Carthat(String modelName, **int** mmdelPrice, **int** modelNumber, **double** d) {

**super**(modelName, mmdelPrice, modelNumber);

**this**.discountRate = d;

}

**public** String display() {

**return** "Carthat [discountRate=" + discountRate + ", modelName=" + modelName + ", mmdelPrice=" + mmdelPrice

+ ", modelNumber=" + modelNumber + "]";

}

**public void** discount()

{

System.***out***.println("Discount Price:"+(mmdelPrice-(discountRate\*mmdelPrice)));

}

}

**public class** Vehicle\_Drive {

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

Carthat obj=**new** Carthat("Duster", 1200000, 7768, 0.2); obj.discount();

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

}

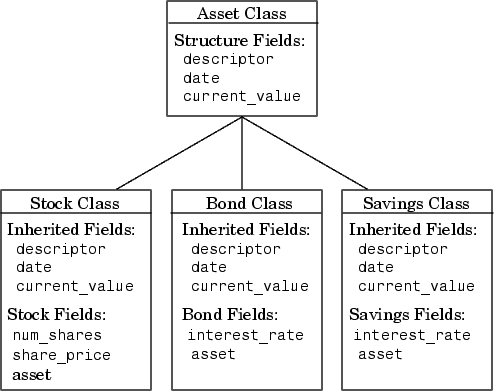
}

Discount Price:960000.0

Carthat [discountRate=0.2, modelName=Duster, mmdelPrice=1200000, modelNumber=7768]

### Implement the below Diagram.

**Here, Asset class is an abstract class containing an abstract method displayDetails() method. Stock, bond and Savings class inherit the Asset class and displayDetails() method is defined in every class.**



**abstract class** Asset{

**protected** String descriptor; **protected** String date; **protected int** currentvalue;

**public** Asset(String descriptor, String date, **int** currentvalue) {

**super**();

**this**.descriptor = descriptor; **this**.date = date; **this**.currentvalue = currentvalue;

}

**abstract public** String display();

}

**class** Stock **extends** Asset{

**int** num\_share; **int** share\_price; **int** asset;

asset) {

**public** Stock(String descriptor, String date, **int** currentvalue, **int** num\_share, **int** share\_price, **int**

**super**(descriptor, date, currentvalue); **this**.num\_share = num\_share; **this**.share\_price = share\_price; **this**.asset = asset;

}

**public** String display() {

**return** "Stock [num\_share=" + num\_share + ", share\_price=" + share\_price + ", asset=" + asset + ", descriptor="

+ descriptor + ", date=" + date + ", currentvalue=" + currentvalue + "]";

}

}

**class** Bond **extends** Asset{

**int** intrest\_rate;

**int** asset;

**public** Bond(String descriptor, String date, **int** currentvalue, **int** intrest\_rate, **int** asset) {

**super**(descriptor, date, currentvalue); **this**.intrest\_rate = intrest\_rate; **this**.asset = asset;

}

**public** String display() {

**return** "Bond [intrest\_rate=" + intrest\_rate + ", asset=" + asset + ", descriptor=" + descriptor + ", date="

+ date + ", currentvalue=" + currentvalue + "]";

}

}

**class** Savings **extends** Asset{

**int** intrest\_rate;

**int** asset;

**public** Savings(String descriptor, String date, **int** currentvalue, **int** intrest\_rate, **int** asset) {

**super**(descriptor, date, currentvalue); **this**.intrest\_rate = intrest\_rate; **this**.asset = asset;

}

**public** String display() {

**return** "Savings [intrest\_rate=" + intrest\_rate + ", asset=" + asset + ", descriptor=" + descriptor + ", date="

+ date + ", currentvalue=" + currentvalue + "]";

}

}

**public class** Abstract\_Asset {

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

Asset obj=**new** Stock("Bruce", "17-08-2019", 50000, 10, 2500, 25000);

Asset obj1=**new** Bond("Barry", "16-08-2019", 40000, 5, 75000);

Asset obj2=**new** Savings("Soumyadip", "15-08-2019", 70000, 3,50000); System.***out***.println(obj.display());

System.***out***.println(obj1.display()); System.***out***.println(obj2.display());

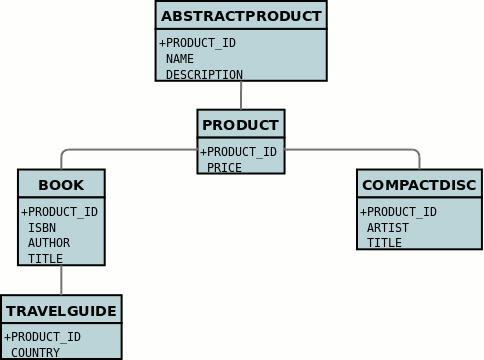
}

}

Stock [num\_share=10, share\_price=2500, asset=25000, descriptor=Bruce, date=17-08-2019, currentvalue=50000]

Bond [intrest\_rate=5, asset=75000, descriptor=Barry, date=16-08-2019, currentvalue=40000] Savings [intrest\_rate=3, asset=50000, descriptor=Soumyadip, date=15-08-2019, currentvalue=70000]

### Implement the below Diagram. Here AbstractProduct is only abstract class.



**abstract class** AbstractProduct{

**protected int** productId; **protected** String name; **protected** String description;

**public** AbstractProduct(**int** productId, String name, String description) {

**super**();

**this**.productId = productId; **this**.name = name; **this**.description = description;

}

}

**class** Product **extends** AbstractProduct{

**protected int** price;

**public** Product(**int** productId, String name, String description, **int** price) {

**super**(productId, name, description);

**this**.price = price;

}

}

**class** Book **extends** Product{

**protected int** ISBN;

**protected** String Author,Title;

**public** Book(**int** productId, String name, String description, **int** price, **int** iSBN, String author, String title) {

**super**(productId, name, description, price); ISBN = iSBN;

Author = author; Title = title;

}

**public** String display() {

**return** "Book [ISBN=" + ISBN + ", Author=" + Author + ", Title=" + Title + ", price=" + price

+ ", productId="

}

}

+ productId + ", name=" + name + ", description=" + description + "]";

**class** Travel\_Guide **extends** Book{

**protected** String Country;

author,

**public** Travel\_Guide(**int** productId, String name, String description, **int** price, **int** iSBN, String

String title, String country) {

**super**(productId, name, description, price, iSBN, author, title); Country = country;

}

**public** String display() {

Title=" + Title

**return** "Travel\_Guide [Country=" + Country + ", ISBN=" + ISBN + ", Author=" + Author + ",

+ ", price=" + price + ", productId=" + productId + ", name=" + name + ",

description=" + description

}

}

+ "]";

**class** CompactDisc **extends** Product{

**protected** String Artist;

**protected** String Title;

title) {

**public** CompactDisc(**int** productId, String name, String description, **int** price, String artist, String

**super**(productId, name, description, price); Artist = artist;

Title = title;

}

**public** String display() {

**return** "CompactDisc [Artist=" + Artist + ", Title=" + Title + ", price=" + price + ", productId=" + productId

+ ", name=" + name + ", description=" + description + "]";

}

}

**public class** Abstract\_Product {

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

Book obj=**new** Book(101,"Chronicles","Chronicles, Volume One is a memoir written by American musician Bob Dylan."

,700,1786,"Bob Dylan","The Chronicles:Vol 1");

Edition"

Travel\_Guide obj1=**new** Travel\_Guide(102,"My Travel Journal","My Travel Journal:1st

,300,1786,"by The Unscripted Life (Author)","My Travel

Journal","India");

Beatles"

CompactDisc obj2=**new** CompactDisc(103,"Abbey Road","Abbey Road:1969/The

,5000,"The Beatles","Abbey Road:1969");

System.***out***.println(obj.display()); System.***out***.println(obj1.display()); System.***out***.println(obj2.display());

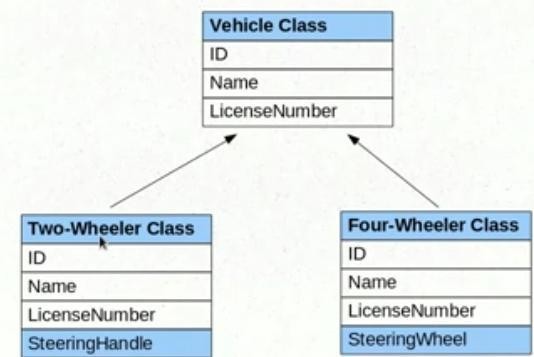
}

}

Book [ISBN=1786, Author=Bob Dylan, Title=The Chronicles:Vol 1, price=700, productId=101, name=Chronicles, description=Chronicles, Volume One is a memoir written by American musician Bob Dylan.]

Travel\_Guide [Country=India, ISBN=1786, Author=by The Unscripted Life (Author), Title=My Travel Journal, price=300, productId=102, name=My Travel Journal, description=My Travel Journal:1st Edition] CompactDisc [Artist=The Beatles, Title=Abbey Road:1969, price=5000, productId=103, name=Abbey Road, description=Abbey Road:1969/The Beatles]

### Implement the below Diagram



**abstract class** Vechicle{

### protected int ID;

**protected** String name;

**protected** String LicenseNumber;

**public** Vechicle(**int** iD, String name, String licenseNumber) { ID = iD;

**this**.name = name; LicenseNumber = licenseNumber;

}

**abstract public** String display();

}

**class** Two\_Wheeler **extends** Vechicle{

**int** SteeringHandle;

**public** Two\_Wheeler(**int** iD, String name, String licenseNumber, **int** steeringHandle) {

**super**(iD, name, licenseNumber); SteeringHandle = steeringHandle;

}

@Override

**public** String display() {

**return** "Two\_Wheeler [SteeringHandle=" + SteeringHandle + ", ID=" + ID + ", name=" + name + ", LicenseNumber="

+ LicenseNumber + "]";

}

}

**class** Four\_Wheeler **extends** Vechicle{

**int** SteeringWheel;

**public** Four\_Wheeler(**int** iD, String name, String licenseNumber, **int** steeringWheel) {

**super**(iD, name, licenseNumber); SteeringWheel = steeringWheel;

}

**public** String display() {

**return** "Four\_Wheeler [SteeringWheel=" + SteeringWheel + ", ID=" + ID + ", name=" + name + ", LicenseNumber="

+ LicenseNumber + "]";

}

}

**public class** Vehicle\_Interface {

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

Vechicle obj=**new** Two\_Wheeler(1011,"KTM","WBIO0897",1); Vechicle obj1=**new** Four\_Wheeler(1011,"Baleno","WBIO4547",1);

System.out.println(obj.display());

System.out.println(obj1.display());

}

}

Two\_Wheeler [SteeringHandle=1, ID=1011, name=KTM, LicenseNumber=WBIO0897] Four\_Wheeler [SteeringWheel=1, ID=1011, name=Baleno, LicenseNumber=WBIO4547]

### Write a program to implement the Multiple Inheritance (Bank Interface, Customer & Account classes).

**interface** Bank{

**final** String ***name***="HDFC";

**public** String details();

}

**class** Customer{

**protected** String CustomerName;

**public** Customer(String name) {

**super**();

**this**.CustomerName = name;

}

}

**class** Account\_Main **extends** Customer **implements** Bank{

**int** acNo;

**public** Account\_Main(String name, **int** acNo) {

**super**(name); **this**.acNo = acNo;

}

**public** String details() {

**return** "Account [acNo=" + acNo + ", CustomerName=" + CustomerName + "]";

}

}

**public class** Bank\_Multiple\_Inherit {

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

Bank obj=**new** Account\_Main("Soumyadip",12990991); System.***out***.print(obj.details());

}

}

Account [acNo=12990991, CustomerName=Soumyadip]

### Write a program to implement the Multiple Inheritance (Gross Interface, Employee & Salary classes).

**interface** Gross{

**public** String details();

}

**class** Employee{

**protected** String EmployeeName;

**public** Employee(String name) {

**super**();

**this**.EmployeeName = name;

}

}

**class** Salary **extends** Employee **implements** Gross{

**int** sal;

**public** Salary(String name, **int** sal) {

**super**(name); **this**.sal = sal;

}

@Override

**public** String details() {

**return** "Salary [Salary=" + sal + ", EmployeeName=" + EmployeeName + "]";

}

}

**public class** Gross\_Multiple\_Interface {

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

Gross obj=**new** Salary("Soumyadip",30000); System.***out***.print(obj.details());

}

}

Salary [sal=30000, EmployeeName=Soumyadip]

### Program to create a interface 'Mango' and implement it in classes 'Winter' and 'Summer'.

**interface** Mango{

**abstract public** String Availability();

}

**class** Summer **implements** Mango{

**boolean** Availability ;

**public** Summer(**boolean** availability) {

Availability = availability;

}

**public** String Availability() {

**return** "Summer [Availability=" + Availability + "]";

}

}

**class** Winter **implements** Mango{

**boolean** Availability ;

**public** Winter(**boolean** availability) {

Availability = availability;

}

**public** String Availability() {

**return** "Winter [Availability=" + Availability + "]";

}

}

**public class** Mango\_Interface {

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

Mango obj=**new** Summer(**true**); Mango obj1=**new** Winter(**false**); System.***out***.println(obj.Availability()); System.***out***.println(obj1.Availability());

}

}

Summer [Availability=true] Winter [Availability=false]

### Program to implement the Multiple Inheritance (Exam Interface, Student & Result classes).

**interface** Exam{

**abstract public** String markSheet();

}

**class** Student\_Exam{

**protected** String name;

**public** Student\_Exam(String name) {

**super**();

**this**.name = name;

}

**public void** get()

{

}

}

**class** Results **extends** Student\_Exam **implements** Exam{

**int** percent;

**public** Results(String name, **int** percent) {

**super**(name); **this**.percent = percent;

}

**public** String markSheet() {

**return** "Result [percentage=" + percent + ", name=" + name + "]";

}

}

**public class** Exam\_Interface {

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

Exam obj=**new** Results("Soumyadip",85); System.***out***.println(obj.markSheet());

}

}

Result [percentage=85, name=Soumyadip]

### Program to demonstrate use of hierarchical inheritance using interface.

**interface** A

{

**public void** displayA();

}

### interface B extends A

{

**public void** displayB();

}

### interface C extends A

{

**public void** displayC();

}

**class** Hierarchy **implements** B,C{

**public void** displayC() {

System.***out***.println("Hi this is C");

}

**public void** displayB() {

System.***out***.println("Hi this is B");

}

**public void** displayA() {

System.***out***.println("Hi this is A");

}

}

**public class** Hierarchical\_Interface {

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

{

Hierarchy obj=**new** Hierarchy(); obj.displayA();

obj.displayB(); obj.displayC();

}

}

Hi this is A Hi this is B Hi this is C

### Java program to Perform Payroll Using Interface (Multiple Inheritance).

**interface** PayRoll{

**public** String totalPay();

}

**class** Employee\_Pay{

**protected** String EmployeeName;

**public** Employee\_Pay(String name) {

**super**();

**this**.EmployeeName = name;

}

}

**class** Salary\_Pay **extends** Employee\_Pay **implements** PayRoll{

**int** sal;

**public** Salary\_Pay(String name, **int** sal) {

**super**(name); **this**.sal = sal;

}

**public** String totalPay() {

**return** "Salary [Salary=" + sal + ", EmployeeName=" + EmployeeName + "]";

}

}

**public class** PayRoll\_Multiple\_Interface2 {

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

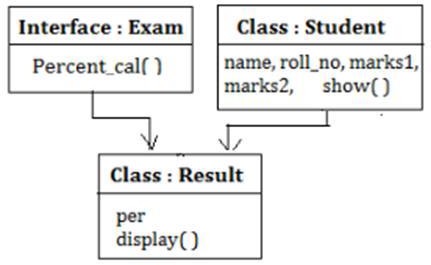
Gross obj=**new** Salary("Soumyadip",50000); System.***out***.print(obj.details());

}

}

Salary [Salary=50000, EmployeeName=Soumyadip]

### Implement the following diagram.



**interface** Exam\_{

**public void** percentCall();

}

**class** Student\_{

**protected** String name; **protected int** roll; **protected int** marks1; **protected int** marks2;

**public** Student\_(String name, **int** roll, **int** marks1, **int** marks2) {

**super**();

**this**.name = name; **this**.roll = roll; **this**.marks1 = marks1; **this**.marks2 = marks2;

}

**public** String show() {

**return** "Student [name=" + name + ", roll=" + roll + ", marks1=" + marks1 + ", marks2=" +

marks2 + "]";

}

}

**class** Result **extends** Student\_ **implements** Exam\_{

**int** per;

**public** Result(String name, **int** roll, **int** marks1, **int** marks2) {

**super**(name, roll, marks1, marks2);

}

**public void** percentCall() {

per=(marks1+marks2)/2; System.***out***.println("Percentage :"+per);

}

**public** String display() {

**return** "Result [percentage=" + per + ", name=" + name + ", roll=" + roll + ", marks1=" + marks1 + ", marks2=" + marks2

+ "]";

}

}

**public class** Result\_Driver {

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

Result obj=**new** Result("Soumyadip",35,85,90); obj.percentCall(); System.***out***.println(obj.show()); System.***out***.println(obj.display());

}

}

Percentage :87

Student [name=Soumyadip, roll=35, marks1=85, marks2=90]

Result [percentage=87, name=Soumyadip, roll=35, marks1=85, marks2=90]

# Week 7

**Objective: Implement the concepts of Exception Handling in Java.**

Assignments:

1. Write a Java program to show the use of all keywords for exception handling public class ALl\_Exception {

public static void main(String[] args) {

int a=50; try{

int sum=a/0;

}

catch(ArithmeticException e)

{

}

finally{

}

}

System.*out*.println("Exception: "+e); System.*out*.println("Executing finally block");

}

Exception: java.lang.ArithmeticException: / by zero Executing finally block

1. Write a Java program using try and catch to generate NegativeArrayIndex Exception and Arithmetic Exception.

public class NegArray\_Arithm\_Exception { public static void main(String[] args) {

int arr[];

try{

}

arr=new int[-10];

catch(NegativeArraySizeException f){ System.*out*.println("Exception "+f);

}

int a=50;

try{

int sum=a/0;

}

catch(ArithmeticException e)

{

System.*out*.println("Exception: "+e);

}

}

}

Exception java.lang.NegativeArraySizeException Exception: java.lang.ArithmeticException: / by zero

1. Define an exception called ¡§NoMatchFoundException¡¨ that is thrown when a string is not equal to

¡§University¡¨. Write a program that uses this exception.

class NoMatchFoundException extends Exception{ NoMatchFoundException(String s){

super(s);

}

}

public class NoMatchFoundException\_Driver {

public static void Match(String z) throws NoMatchFoundException

{

if(z!="¡§University¡¨")

{

throw new NoMatchFoundException("NoMatchFoundException Gen");

}

}

public static void main(String args[])

{

String z="abc"; try {

*Match*(z);

} catch (Exception e) { System.*out*.println(e);

}

System.*out*.print("Rest Code");

}

}

NoMatchFoundException: NoMatchFoundException Gen Rest Code

1. Write a class that keeps a running total of all characters passed to it (one at a time) and throws an exception if it is passed a non-alphabetic character.

class NonAlphabeticException extends Exception{ NonAlphabeticException(String s){

super(s);

}

}

class Alpha{

public static void alpha(char a) throws NonAlphabeticException

{

if(a<97 || a>122)

{

throw new NonAlphabeticException("NonAlphabeticException");

}

}

}

public class Alphabetic\_exception {

public static void main(String args[])

{

char[] a={'a','b','c','1'}; int i=0; while(i<a.length)

{

try {

Alpha.*alpha*(a[i]);

} catch (NonAlphabeticException e) { System.*out*.println(e);

} i++;

}

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

}

}

NonAlphabeticException: NonAlphabeticException Rest

1. Write a program called Factorial.java that computes factorials and catches the result in an array of type long for reuse. The long type of variable has its own range. For example 20! Is as high as the range of long type. So check the argument passes and ¡§throw an exception¡¨, if it is too big or too small.

„h If x is less than 0 throw an IllegalArgumentException with a message ¡§Value of x must be positive¡¨.

„h If x is above the length of the array throw an IllegalArgumentException with a message ¡§Result will

overflow¡¨.

Here x is the value for which we want to find the factorial.

public class Factorial\_Exception {

public static void calculateFactorial(long n) { if (n < 0)

throw new IllegalArgumentException("n must be positive"); else if (n > 20)

throw new IllegalArgumentException("n must be < 20"); else

{

long z=n; long fact=1; while(n>1)

{

fact\*=n; n--;

}

System.*out*.println("Factorial of "+z+" = "+fact);

}

}

public static void main(String args[])

{

*calculateFactorial*(5); *calculateFactorial*(21);

}

}

Factorial of 5 = 120 java.lang.IllegalArgumentException: n must be < 20

1. Write a class that keeps a running total of all characters passed to it (one at a time) and throws an exception if it is passed a non-alphabetic character.

class NonAlphabeticException extends Exception{ NonAlphabeticException(String s){

super(s);

}

}

class Alpha{

public static void alpha(char a) throws NonAlphabeticException

{

if(a<97 || a>122)

{

throw new NonAlphabeticException("NonAlphabeticException");

}

}

}

public class Alphabetic\_exception {

public static void main(String args[])

{

char[] a={'a','b','c','1'}; int i=0; while(i<a.length)

{

try {

Alpha.*alpha*(a[i]);

} catch (NonAlphabeticException e) { System.*out*.println(e);

} i++;

}

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

}

}

NonAlphabeticException: NonAlphabeticException Rest

1. Write a program that outputs the name of the capital of the country entered at the command line. The program should throw a ¡§NoMatchFoundException¡¨ when it fails to print the capital of the country entered at the command line.

import java.util.Scanner;

class NoMatchFoundException extends Exception{ NoMatchFoundException(String s){

super(s);

}

}

public class CapitalException\_Driver {

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter teh capital name"); String str=sc.next();

String cap="delhi"; try {

if(str.equals(cap))

{

}

else

{

}

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

throw new NoMatchFoundException("NoMatchFoundException");

} catch (Exception e) {

System.*out*.println(e);

}

System.*out*.print("Rest Code");

}

}

Enter teh capital name Chennai

NoMatchFoundException: NoMatchFoundException Rest Code

1. Write a program that takes a value at the command line for which factorial is to be computed. The program must convert the string to its integer equivalent. There are three possible user input errors that can prevent the program from executing normally.

„h The first error is when the user provides no argument while executing the program and an

ArrayIndexOutOfBoundsException is raised. You must write a catch block for this.

„h The second error is NumberFormatException that is raised in case the user provides a non-integer (float double) value at the command line.

„h The third error is IllegalArgumentException. This needs to be thrown manually if the value at the

command line is 0.

import java.util.Scanner;

public class Factorial\_Three\_Exception { public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); String str=sc.next();

int n=0; try{

n=Integer.*parseInt*(str); if(n==0)

throw new IllegalArgumentException();

int z=n; int fact=1;

while(n>1)

{

fact\*=n; n--;

}

System.*out*.println("Factorial of "+z+" = "+fact);

} catch(ArrayIndexOutOfBoundsException e)

{

System.*out*.print(e);

}

catch(NumberFormatException e)

{

System.*out*.print(e);

}

catch(IllegalArgumentException e)

{

System.*out*.print(e);

}

}

}

1.00

java.lang.NumberFormatException: For input string: "1.00"

0

java.lang.IllegalArgumentException

5

Factorial of 5 = 120

1. Create a user-defined exception named CheckArgument to check the number of arguments passed through the command line. If the number of argument is less than 5, throw the CheckArgumentexception, else print the addition of all the five numbers.

import java.util.Scanner;

class CheckArgumentexception extends Exception{ CheckArgumentexception(String s){

super(s);

}

}

public class CheckArgumentexception\_Driver {

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("enter the limit"); int n=sc.nextInt();

System.*out*.println("enter "+n+" numbers"); int[] arr=new int[n];

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

arr[i]=sc.nextInt();

try

{

if(n<5) else

{

throw new CheckArgumentexception("n is less than 5"); int sum=0;

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

sum+=arr[i];

System.*out*.println("Sum is:"+sum);

}

}

catch(Exception e)

{

System.*out*.print(e);

}

}

}

enter the limit 3

enter 3 numbers

1

2

3

CheckArgumentexception: n is less than 5

1. Consider a Student examination database system that prints the mark sheet of students. Input the following from the command line.
2. Student¡¦s Name
3. Marks in six subjects

These marks should be between 0 to 50. If the marks are not in the specified range, raise a RangeException, else find the total marks and prints the percentage of the students.

import java.util.Scanner;

class RangeException extends Exception{ RangeException(){

super("Marks should be between 0 to 50");

}

}

public class RangeException\_Student {

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter the Student name"); String str=sc.next();

System.*out*.println("enter "+6+" marks");

int[] arr=new int[6]; for(int i=0;i<6;i++)

arr[i]=sc.nextInt();

try

{

int sum=0;

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

{

if(arr[i]<0 || arr[i]>50)

throw new RangeException(); sum+=arr[i];

}

System.*out*.print("Percentage is : "+(sum/6)\*2);

}

catch(RangeException e)

{

System.*out*.print(e);

}

}

}

Enter the Student name Soumyadip

enter 6 marks

45

45

45

900

45

40

RangeException: Marks should be between 0 to 50

Enter the Student name Soumyadip

enter 6 marks

40

40

40

40

40

40

Percentage is : 80

1. Write a java program to create an custom Exception that would handle at least 2 kind of Arithmetic Exceptions while calculating a given equation (e.g. X+Y\*(P/Q)Z-I)

import java.util.Scanner;

class CustomArithmeticException extends Exception{ CustomArithmeticException(String s){

super(s);

}

}

public class Two\_ArithmeticException {

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); int x,y,p,q,z,l; System.*out*.println("Enter X"); x=sc.nextInt(); System.*out*.println("Enter Y"); y=sc.nextInt(); System.*out*.println("Enter P"); p=sc.nextInt(); System.*out*.println("Enter P"); q=sc.nextInt(); System.*out*.println("Enter Z"); z=sc.nextInt(); System.*out*.println("Enter L"); l=sc.nextInt();

try

{

if(q==0)

throw new CustomArithmeticException("Cannot divided by 0");

int sum=(x+y\*(p/q)\*z-l); if(sum<0)

throw new CustomArithmeticException("Sum cannot be negetive"); System.*out*.print(sum);

}

catch(CustomArithmeticException e)

{

System.*out*.print(e);

}

}

}

Enter X 10

Enter Y 20

Enter P 30

Enter P 0

Enter Z 40

Enter L 50

CustomArithmeticException: Cannot divided by 0

Enter X 1

Enter Y 2

Enter P 3

Enter P 4

Enter Z 5

Enter L 6

CustomArithmeticException: Sum cannot be negative

1. Create two user-defined exceptions named ¡§TooHot¡¨ and ¡§TooCold¡¨ to check the temperature (in Celsius) given by the user passed through the command line is too hot or too cold.

If temperature > 35, throw exception ¡§TooHot¡¨. If temperature <5, throw exception ¡§TooCold¡¨.

Otherwise, print ¡§Normal¡¨ and convert it to Farenheit.

import java.util.Scanner;

class TooHot extends Exception{ TooHot(){

super("Temperature is too-hot");

}

}

class TooCold extends Exception{

TooCold(){

super("Temperature is too-cold");

}

}

public class TempException {

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); int x,y,p,q,z,l;

System.*out*.println("Enter the temperature (Celsius):"); x=sc.nextInt();

try

{

if(x<0) if(x>35)

throw new TooCold(); throw new TooHot();

System.*out*.println("Temperature is (Celsius):"+x); System.*out*.print("Temperature is (Fahrenheit ):"+(x\*(9/5)+32));

}

catch(TooHot e)

{

System.*out*.print(e);

}

catch(TooCold e)

{

System.*out*.print(e);

}

}

}

Enter the temperature (Celsius):

-3

TooCold: Temperature is too-cold Enter the temperature (Celsius):

50

TooHot: Temperature is too-hot Enter the temperature (Celsius):

19

Temperature is (Celsius):19 Temperature is (Fahrenheit ):51

1. Consider an Employee recruitment system that prints the candidate name based on the age criteria. The name and age of the candidate are taken as Input.Create two user-defined exceptions named

¡§TooOlder¡¨ and ¡§TooYounger¡¨

„h If age>45, throw exception ¡§TooOlder¡¨.

„h If age<20, throw exception ¡§TooYounger¡¨.

„h Otherwise, print ¡§Eligible¡¨ and print the name of the candidate.

import java.util.Scanner;

class TooYounger extends Exception{ TooYounger(){

super("TooYounger");

}

}

class TooOlder extends Exception{

TooOlder(){

super("TooOlder");

}

}

public class AgeException {

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter the Name:"); String name=sc.next();

System.*out*.println("Enter the age:"); int x=sc.nextInt();

try

{

if(x<20) if(x>40)

throw new TooYounger(); throw new TooOlder();

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

}

catch(TooYounger e)

{

System.*out*.print(e);

}

catch(TooOlder e)

{

System.*out*.print(e);

}

}

}

Enter the Name:

Tom

Enter the age:

19

TooYounger: TooYounger

Enter the Name:

Bruce

Enter the age:

52

TooOlder: TooOlder

Enter the Name:

Soumyadip Enter the age:

21

(Eligible) Name :Soumyadip

1. Consider a ¡§Binary to Decimal¡¨ Number conversion system which only accepts binary number as Input. If user provides a decimal number a custom Exception ¡§WrongNumberFormat¡¨ exception will be thrown. Otherwise, it will convert into decimal and print into the screen.

import java.util.Scanner;

class WorngNumberException extends Exception{ WorngNumberException(){

super("Please enter the binary number");

}

}

public class Binary\_Decimal\_Exception {

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter the number (in binary):"); String str=sc.next();

try

{

}

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

if(str.charAt(i)!='0' && str.charAt(i)!='1')

throw new WorngNumberException();

int a=Integer.*parseInt*(str,2); System.*out*.print("Decimal :"+a);

catch(Exception e)

{

System.*out*.print(e);

}

}

}

Enter the number (in binary):

121

WorngNumberException: Please enter the binary number

Enter the number (in binary):

101

Decimal :5

1. Write a Java Program that Implement the Nested Try Statements.

public class NestedTry {

public static void main(String args[])

{

try {

int a[] = { 1,2,3 };

System.*out*.println(a[10]); try {

int x = a[2] / 0;

}

catch (ArithmeticException f) { System.*out*.println(f);

}

}

catch (ArrayIndexOutOfBoundsException e) { System.*out*.println(e);

}

}

}

java.lang.ArithmeticException: / by zero

1. Write a Java Program to Create Account with 500 Rs Minimum Balance, Deposit Amount, Withdraw Amount and Also Throws LessBalanceException.

Java Program Which has a Class Called LessBalanceException Which returns the Statement that Says WithDraw Amount(\_Rs) is Not Valid

Java Program that has a Class Which Creates 2 Accounts, Both Account Deposit Money and One Account Tries to WithDraw more Money Which Generates a LessBalanceException Take Appropriate Action for the Same.

import java.util.Scanner;

class LessBalanceException extends Exception{ LessBalanceException(int i){

super("WithDraw Amount "+i+" is not valid");

}

}

public class Bank\_Exception {

public static void main(String[] args) { int am=500;

System.*out*.println("Account created with initial balance 500");

int x=0;

Scanner sc=new Scanner(System.*in*); while(x!=1)

{

System.*out*.println("1. Diposit"); System.*out*.println("2. Withdraw"); int a=sc.nextInt();

if(a==1)

{

System.*out*.println("Enter the ammount"); int z=sc.nextInt();

am+=z;

System.*out*.println("Current balance :"+am);

}

else if(a==2)

{

System.*out*.println("Enter the ammount");

int z=sc.nextInt(); if(z>am)

try {

throw new LessBalanceException(z);

} catch (LessBalanceException e) { System.*out*.println(e);

else

}

am-=z;

}

else

{

}

}

}

System.*out*.println("Current balance :"+am);

x=1;

}

Account created with initial balance 500

1. Diposit
2. Withdraw 1

Enter the ammount 5000

Current balance :5500

1. Diposit
2. Withdraw 2

Enter the ammount 2000

Current balance :3500

1. Diposit
2. Withdraw 2

Enter the ammount 10000

LessBalanceException: WithDraw Amount 10000 is not valid Current balance :3500

1. Consider a Library Management System, where a user wants to find a book. If the book is present in Library (Hint: Use predefined array), then it will print the book. Otherwise it will throw an exception

¡§BookNotFoundException¡¨.

import java.awt.print.Book; import java.util.Scanner;

class BookNotFoundException extends Exception{

BookNotFoundException(){

super("Book not found exception");

}

}

public class Library\_Exception\_Driver {

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*); int books[]={101,202,303,404,505};

System.*out*.println("Enter the book no.:"); int key=sc.nextInt();

int flag=0;

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

{

}

try

{

if(books[i]==key)

flag=1;

if(flag==0)

throw new BookNotFoundException();

else

System.*out*.print("Book found");

}

catch(BookNotFoundException e)

{

System.*out*.print(e);

}

}

}

Enter the book no.:

1010

BookNotFoundException: Book not found exception

Enter the book no.: 101

Book found

1. Consider a Quiz Management System, where a user needs to answer 5 questions. If any of the answer is wrong, throw an exception ¡§NotCorrectException¡¨. If the answer is correct give a message ¡§good! The answer is correct¡¨.

import java.util.Scanner;

class NotCorrectException extends Exception{ NotCorrectException(){

super("Wrong answer");

}

}

public class Quiz\_Exception {

public static void main(String arrg[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Captain of Indian Cricket team?"); System.*out*.println("1: Virat Kohli"); System.*out*.println("2: MS. Dhoni");

if(sc.nextInt()==1)

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

else

try {

throw new NotCorrectException();

} catch (NotCorrectException e) { System.*out*.println(e);

}

System.*out*.println("EX Prime Minister of India?"); System.*out*.println("1: Dr. Manmohan Singh"); System.*out*.println("2: Arvind Kejriwal"); if(sc.nextInt()==1)

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

else

try {

throw new NotCorrectException();

} catch (NotCorrectException e) { System.*out*.println(e);

}

System.*out*.println("Who wrote the C language?"); System.*out*.println("1: Bill Gates");

System.*out*.println("2: Dennis Ritchie"); if(sc.nextInt()==2)

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

else

try {

throw new NotCorrectException();

} catch (NotCorrectException e) { System.*out*.println(e);

}

System.*out*.println("JAVA is written in which year?"); System.*out*.println("1: 1992");

System.*out*.println("2: 1995"); if(sc.nextInt()==2)

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

else

try {

throw new NotCorrectException();

} catch (NotCorrectException e) { System.*out*.println(e);

}

System.*out*.println("When was TCS established?"); System.*out*.println("1: 1 April 1968");

System.*out*.println("2: 1 April 1972"); if(sc.nextInt()==1)

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

else

try {

throw new NotCorrectException();

} catch (NotCorrectException e) { System.*out*.println(e);

}

}

}

Captain of Indian Cricket team? 1: Virat Kohli

2: MS. Dhoni

2

NotCorrectException: Wrong answer EX Prime Minister of India?

1: Dr. Manmohan Singh 2: Arvind Kejriwal

2

NotCorrectException: Wrong answer

Who wrote the C language? 1: Bill Gates

2: Dennis Ritchie

2

Good

JAVA is written in which year? 1: 1992

2: 1995

2

Good

When was TCS established? 1: 1 April 1968

2: 1 April 1972

2

NotCorrectException: Wrong answer

1. Write a program to raise a user defined exception if username is less than 6 characters and password does not match.

import java.util.Scanner;

class PasswordException extends Exception{ PasswordException(String s){

super(s);

}

}

public class Password\_Exception {

public static void main(String arrg[])

{

String pass="UEMKCSEJAVA"; Scanner sc=new Scanner(System.*in*);

System.*out*.print("Enter the username :"); String user1=sc.next(); if(user1.length()<6)

try {

throw new PasswordException("Length is less than 6");

} catch (PasswordException e) { System.*out*.print(e);

}

System.*out*.print("Enter the password :"); String str=sc.next();

if(str.equals(pass))

else

try {

System.*out*.print("Password Matched");

throw new PasswordException("Wrong Password");

} catch (PasswordException e) { System.*out*.print(e);

}

}

}

Enter the username :CSE PasswordException: Length is less than 6

Enter the username :Soumya Enter the password :UEMKCSE

PasswordException: Wrong Password

Enter the username :Soumya

Enter the password :UEMKCSEJAVA Password Matched

1. Write a program to accept a password from the user and throw 'Authentication Failure' exception if the password is incorrect.

import java.util.Scanner;

class AuthenticationFailure extends Exception{ AuthenticationFailure(){

super("AuthenticationFailure");

}

}

public class USER\_PASS\_Exception {

public static void main(String arrg[])

{

String pass="UEMKCSEJAVA"; Scanner sc=new Scanner(System.*in*);

System.*out*.println("Enter the password :"); String str=sc.next();

if(str.equals(pass))

System.*out*.println("Password Matched");

else

try {

throw new AuthenticationFailure();

} catch (AuthenticationFailure e) { System.*out*.print(e);

}

}

}

Enter the password : UEMKCSEJAVA Password Matched

Enter the password : UEMKCSE AuthenticationFailure: AuthenticationFailure

1. Write a program to input name and age of a person and throw a user-defined exception, if the entered age is negative.

import java.util.Scanner;

class NegAgeException extends Exception{ NegAgeException(){

super("Age should'nt be negetive");

}

}

public class Age\_Name\_Exception {

public static void main(String arrg[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.print("Enter the name :"); String user1=sc.next();

System.*out*.println("Enter the age :"); int age=sc.nextInt();

if(age<0)

try {

throw new NegAgeException();

} catch (NegAgeException e) { System.*out*.print(e);

}

}

}

Enter the name :Soumyadip Enter the age :

-20

NegAgeException: Age should'nt be negetive

1. Write a program to throw user defined exception if the given number is not positive.

import java.util.Scanner;

class NotPositiveException extends Exception{ NotPositiveException(){

super("Age should'nt be negetive");

}

}

public class NotPositive\_Exception{

public static void main(String arrg[])

{

Scanner sc=new Scanner(System.*in*); System.*out*.println("Enter a number :"); int age=sc.nextInt();

if(age<0)

try {

throw new NotPositiveException();

} catch (NotPositiveException e) { System.*out*.print(e);

}

}

}

Enter a number :

-1

NotPositive\_Exception: Number should'nt be negetive

1. Write a program to throw a user-defined exception "String Mismatch Exception", if two strings are not equal (ignore the case).

import java.util.Scanner;

class StringMismatchException extends Exception{

StringMismatchException (){ super("StringMismatchException ");

}

}

public class String\_Equal\_Exception {

public static void main(String arrg[])

{

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

System.*out*.println("Enter 1st String :"); String str=sc.next(); System.*out*.println("Enter 2nd String :"); String str1=sc.next();

if(str.equals(str1))

System.*out*.println("String Matched");

else

try {

throw new StringMismatchException();

} catch (StringMismatchException e) { System.*out*.print(e);

}

}

}

Enter 1st String :

UEMK

Enter 2nd String :

UEMK

String Matched Enter 1st String :

UEMK

Enter 2nd String :

UEMKJ

StringMismatchException: StringMismatchException

1. Design a stack class. Provide your own stack exceptions namely push exception and pop exception, which throw exceptions when the stack is full and when the stack is empty respectively. Show the usage of these exceptions in handling a stack object in the main.

class StackException extends Exception{ public StackException()

{

super("Stack size exception");

}

}

class Stack\_{

static int *arr*[]=new int[5]; static int *max*=*arr*.length; static int *top*=-1;

static void push(int a)

{

*top*++; if(*top*<*max*)

{

*arr*[*top*]=a; System.*out*.println(a+" pushed");

}

else

{

*top*--; try {

throw new StackException();

} catch (StackException e) { System.*out*.println(e);

}

}

}

static void pop()

{

if(*top*>=0)

System.*out*.println(*arr*[*top*--]+" Popped");

else

{

*top*++; try {

throw new StackException();

} catch (StackException e) { System.*out*.println(e);

}

}

}

}

public class Stack\_Exception {

public static void main(String[] args) { Stack\_.*push*(10);

Stack\_.*push*(20); Stack\_.*push*(30); Stack\_.*push*(40); Stack\_.*push*(50); Stack\_.*push*(50); Stack\_.*pop*();

Stack\_.*pop*();

Stack\_.*pop*();

Stack\_.*pop*();

Stack\_.*pop*();

Stack\_.*pop*();

}

}

10 pushed

20 pushed

30 pushed

40 pushed

50 pushed

StackException: Stack size exception 50 Popped

40 Popped

30 Popped

20 Popped

10 Popped

StackException: Stack size exception

1. Write an application that displays a series of at least five student ID numbers (that you have stored in an array) and asks the user to enter a numeric test score for the student. Create a ScoreException class, and throw a ScoreException for the class if the user does not enter a valid score (zero to 100). Catch the ScoreException and then display an appropriate message. In addition, store a 0 for the student¡¦s score. At the end of the application, display all the student IDs and scores.

import java.util.Scanner;

class ScoreException extends Exception{ ScoreException (){

super("Invaid Score");

}

}

public class Student\_Exception{

public static void main(String arrg[])

{

int enroll[]={101,102,103,104,105};

int marks[]=new int[5];

Scanner sc=new Scanner(System.*in*); for(int i=0;i<enroll.length;i++)

{

System.*out*.println("Enter marks of Enrollment no:"+enroll[i]); int a=sc.nextInt();

if(a<0 || a>100)

{

try {

throw new ScoreException();

} catch (ScoreException e) { System.*out*.println(e);

}

else

{

}

}

}

marks[i]=0;

marks[i]=a;

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

{

System.*out*.println("Enrollment no:"+enroll[i]+" Marks:"+marks[i]);

}

}

}

Enter marks of Enrollment no:101 99

Enter marks of Enrollment no:102

-1

ScoreException: Invaid Score Enter marks of Enrollment no:103 95

Enter marks of Enrollment no:104 101

ScoreException: Invaid Score Enter marks of Enrollment no:105 89

Enrollment no:101 Marks:99 Enrollment no:102 Marks:0 Enrollment no:103 Marks:95 Enrollment no:104 Marks:0 Enrollment no:105 Marks:89

1. Write a GUI based program in Java that will have two buttons viz., RED and GREEN. If user clicks on the RED button the background of the GUI will be painted in red and if user clicks on the GREEN button the background of the GUI will be painted in green.

<applet code ="Buttons" width=200

height=200>

</applet>

import java.awt.\*; import java.awt.event.\*; import java.applet.\*;

public class Buttons extends Applet implements ActionListener

{

Button b1,b2;

public void init()

{

b1=new Button("RED"); add(b1); b1.addActionListener(this);

b2=new Button("GREEN"); add(b2); b2.addActionListener(this);

}

public void actionPerformed(ActionEvent e1)

{

if(e1.getSource()==b1)

{

setBackground(Color.red);

}

if(e1.getSource()==b2)

{

setBackground(Color.green);

}

}

}

1. Design a Java applet that will display “Hello Applet” message in the applet window and set the

background and foreground colour

/\*

<applet code="MyApplet " width=200 height=200>

</applet>

\*/

import java.applet.\*; import java.awt.Color;

import java.awt.\*;

public class MyApplet extends Applet { public void init()

{

setBackground(Color.cyan); setForeground(Color.red);

}

public void paint(Graphics g) {

g.drawString("Hello applet!", 50, 25);

}

}

1. Write a program to show the lifecycle of an Applet.

*/\*<applet code="AppletLifeCycle.class" width="350" height="150"> </applet>\*/ import* [*java*](http://ecomputernotes.com/java/what-is-java/what-is-java-explain-basic-features-of-java-language)*.applet.Applet;*

*import java.awt.Graphics; import java.awt.\*;*

*public class AppletLifeCycle extends Applet*

*{*

*public void init()*

*{*

*setBackground(Color.CYAN); System.out.println("init() called");*

*}*

*public void start(){ System.out.println("Start() called"); }*

*public void paint(Graphics g){ System.out.println("Paint(() called"); }*

*public void stop() { System.out.println("Stop() Called"); }*

*public void destroy() { System.out.println("Destroy)() Called"); }*

*}*

1. Write an applet to draw the following shapes:
   1. Rectangle
   2. Square
   3. Circle.

Import java.awt.\*;

import java.applet.Applet;

public class MyGraphics extends Applet

{

public void paint (Graphics g)

{

// Rectangle g.drawRect(10,10,50,100);

// Square g.drawRect(100,100,50,50);

// Circle

g.drawOval(150, 50, 90, 90);

}

}

# Week 10-Multithreading

//Ascending Descending Thread

class AscOrder extends Thread{

public void run(){

for(int i=1;i<=1000;i++){

System.out.println("Values Of i in Ascending Order"+i);

}

System.out.println("Exiting ascending order...");

}//run exit

}

class DescOrder extends Thread{

public void run(){

for(int j=1000;j>=1;j--){

System.out.println("Values Of j in Descending Order"+j);

}

System.out.println("Exiting descending order...");

}//run exit

}

class AscDesc{

public static void main(String arg[]){

Thread t1=new AscOrder();

t1.start();

Thread t2=new DescOrder();

t2.start();

System.out.println("Exiting Main...");

}

}

//Prime/Non-prime Thread

class Prime extends Thread{

public void run(){

for(int i=2;i<=1000;i++){

int flag=1;

for(int j=2;j<=i/2;j++){

if(i%j==0){

flag=0;

break;

}

}

if(flag==1)

System.out.println("Prime NO.="+i);

}

System.out.println("Exiting Prime...");

}//run exit

}

class NonPrime extends Thread{

public void run(){

for(int i=2;i<=1000;i++){

int flag=1;

for(int j=2;j<=i/2;j++){

if(i%j==0){

flag=0;

break;

}

}

if(flag==0)

System.out.println("NON Prime NO.="+i);

}

System.out.println("Exiting Non Prime...");

}//run exit

}

class PrimeNon{

public static void main(String arg[]){

Thread t1=new Prime();

Thread t2=new NonPrime();

t1.start();

t2.start();

System.out.println("Exiting Main...");

}

}

//Setting priorities(Thread)

/\* Setting priority to threads \*/

class threadA extends Thread{

public void run() {

System.out.println("Start Thread A ....");

for(int i = 5; i >= 1; i--) {

try{

Thread.sleep(1000);

System.out.println("From Thread A: i = "+ i);

}catch(InterruptedException ae){

System.out.println(ae);

}

}

System.out.println("... Exit Thread A");

}

}

class threadB extends Thread{

public void run() {

System.out.println("Start Thread B ....");

for(int j = 5; j >=1; j--) {

try{

Thread.sleep(1000);

System.out.println("From Thread B: j = "+ j);

}catch(InterruptedException ae){

System.out.println(ae);

}

}

System.out.println("... Exit Thread B");

}

}

class threadC extends Thread{

public void run() {

System.out.println("Start Thread C ....");

for(int k = 5; k>= 1; k--) {

try{

Thread.sleep(1000);

System.out.println("From Thread C: k = "+k);

}catch(InterruptedException ae){

System.out.println(ae);

}

}

System.out.println("... Exit Thread C");

}

}

class threadD extends Thread{

public void run(){

System.out.println("From D Thread...");

for(int p=5;p>=1;p--){

try{

Thread.sleep(1000);

System.out.println("From Thread D: p = "+ p);

}catch(InterruptedException ae){

System.out.println(ae);

}

}

System.out.println("Exit.. D Thread..");

}

}

class ThreadPriorityTest{

public static void main (String args[]) {

threadA t1 = new threadA();

threadB t2 = new threadB();

threadC t3 = new threadC();

threadD t4 = new threadD();

t1.setPriority(Thread.MIN\_PRIORITY);

System.out.println(t1.getPriority());

t2.setPriority(t1.getPriority()+2);

System.out.println(t2.getPriority());

t3.setPriority(t2.getPriority() + 2);

System.out.println(t3.getPriority());

t4.setPriority(t3.getPriority()+2);

System.out.println(t4.getPriority());

t1.start(); t2.start(); t3.start();t4.start();

System.out.println("... End of executuion ");

}

}

//Odd Even Thread

class EvenThread extends Thread{

public void run(){

for(int i=0;i<1000;i++){

if(i%2==0){

try{

Thread.sleep(500);

}catch(InterruptedException e){

System.out.println(e);

}

System.out.println("Value of i in EVEVN THREAD"+i);

}

}

System.out.println("EvenThread exit...");

}

}

class OddThread extends Thread{

public void run(){

for(int i=0;i<1000;i++){

if(i%2!=0){

try{

Thread.sleep(500);

}catch(InterruptedException e){

System.out.println(e);

}

System.out.println("Value of i in ODD THREAD"+i);

}

}

System.out.println("OddThread exit...");

}

}

class OddEvenThread{

public static void main(String arg[]){

EvenThread a=new EvenThread();

a.start();

OddThread b=new OddThread();

b.start();

System.out.println("Exiting Main...");

}

}

//Using Runnable(Thread)

class EvenThread implements Runnable{

public void run(){

for(int i=0;i<1000;i++){

if(i%2==0){

try{

Thread.sleep(500);

}catch(InterruptedException e){

System.out.println(e);

}

System.out.println("Value of i in EVEVN THREAD"+i);

}

}

System.out.println("EvenThread exit...");

}

}

class OddThread implements Runnable{

public void run(){

for(int i=0;i<1000;i++){

if(i%2!=0){

try{

Thread.sleep(500);

}catch(InterruptedException e){

System.out.println(e);

}

System.out.println("Value of i in ODD THREAD"+i);

}

}

System.out.println("OddThread exit...");

}

}

class OddEvenThread{

public static void main(String arg[]){

EvenThread a=new EvenThread();

Thread t1= new Thread(a);

t1.start();

OddThread b=new OddThread();

Thread t2=new Thread(b);

t2.start();

System.out.println("Exiting Main...");

}

}