NAME: SAGNIK DAS **ROLL NO:** 55

ENROLLMENT NO: 12023006015037 **SEC:** A **Semester:** 2nd

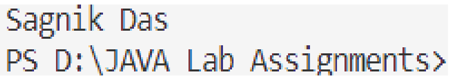
DEPARTMENT: M.C.A.

Week 1

1.Write a Java program to print your name.

Code:

```
public class name
    { public static void main(String args[])
    {  System.out.println("Sagnik Das");
    }
}
```

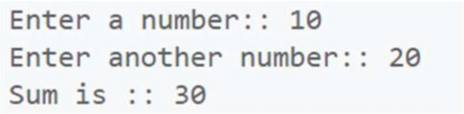


```
Sagnik Das
PS D:\JAVA Lab Assignments>
```

2. Write a Java program to add two numbers.

Code:

```
import java.util.*;
public class add
        {public static void main(String[] args)
        {Scanner reader = new Scanner(System.in);
         System.out.print("Enter a number:: ");
         int num1 = reader.nextInt();
         System.out.print("Enter another number:: ");
         int num2 = reader.nextInt();
         int sum = num1+num2;
         System.out.println("Sum is :: "+sum);
             }
         }
```

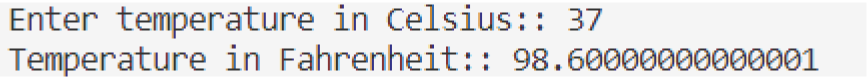


```
Enter a number:: 10
Enter another number:: 20
Sum is :: 30
```

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

Code:

```
import java.util.*;
public class as1_3 {
        public static void main(String[] args) {
        Scanner reader = new Scanner(System.in);
        System.out.print("Enter temperature in Celsius:: ");
        double temp = reader.nextDouble();
        double fer = (temp*1.8)+32;
        System.out.println("Temperature in Fahrenheit:: "+fer);
        }
        }
```

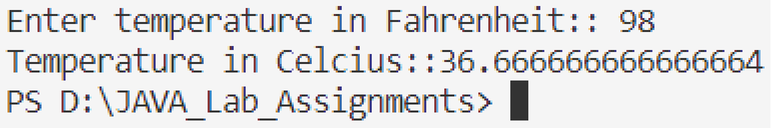


```
Enter temperature in Celsius:: 37
Temperature in Fahrenheit:: 98.60000000000001
```

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

Code: import java.util.*;

```
public class as1_4 {
public static void main(String[] args) {
Scanner reader = new Scanner(System.in);
System.out.print("Enter temperature in Fahrenheit:: ");
double temp = reader.nextDouble();
double cel = (temp-32)*5/9;
System.out.print("Temperature in Celcius::"+cel);
}
}
```

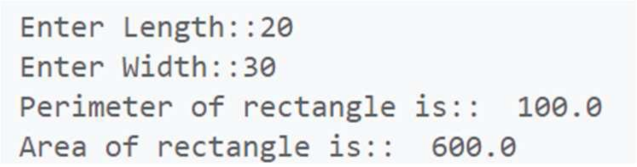


```
Enter temperature in Fahrenheit:: 98
Temperature in Celcius::36.666666666666664
PS D:\JAVA_Lab_Assignments> █
```

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

Code: import java.util.*;

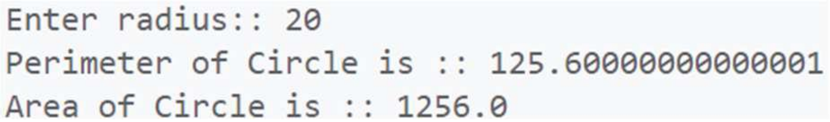
```
public class rect {
public static void main(String[] args) {
Scanner reader = new Scanner(System.in);
System.out.print("Enter Length::");
int len = reader.nextInt();
System.out.print("Enter Width::");
int wid = reader.nextInt();
double peri = 2*(len+wid);
double area = len*wid;
System.out.println("Perimeter of rectangle is:: "+peri);
System.out.println("Area of rectangle is:: "+area);
}
}
```



```
Enter Length::20
Enter Width::30
Perimeter of rectangle is::  100.0
Area of rectangle is::  600.0
```

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

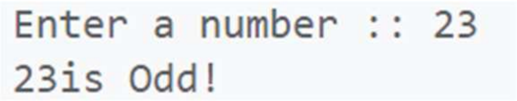
Code:
```
import java.util.*;

public class circle {

public static void main(String[] args) {

Scanner reader = new Scanner(System.in);

System.out.print("Enter radius:: ");

double r = reader.nextDouble();

double peri = 2*3.14*r;

double area = 3.14*r*r;

System.out.println("Perimeter of Circle is :: " +peri);

System.out.println("Area of Circle is :: " +area);

}

}
```



```
Enter radius:: 20
Perimeter of Circle is :: 125.60000000000001
Area of Circle is :: 1256.0
```

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

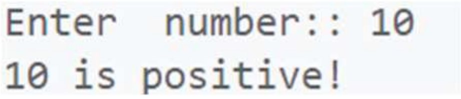
Code:
```
import java.util.*;

public class as1_7 {

public static void main(String[] args) {

Scanner reader = new Scanner(System.in);

System.out.print("Enter a number :: ");

int num1 = reader.nextInt();

if (num1 % 2 == 0){

System.out.println(num1 + " is even!" );

}else{

System.out.println(num1 + "is Odd!");

}

}

}
```



```
Enter a number :: 23
23is Odd!
```

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

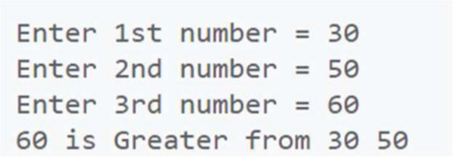
Code:
```
import java.util.*;
class as1_8 {
public static void main(String[] args) {
Scanner read = new Scanner(System.in);
System.out.print("Enter number:: ");
int num1 = read.nextInt();
if (num1<0){
System.out.println(num1 + " is negative!");
}else if(num1 == 0){
System.out.println(num1 + " is Zero!");
}else {
System.out.println(num1 + " is positive!");
}
}
}
```



```
Enter  number:: 10
10 is positive!
```

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

Code:
```
import java.util.*;
public class as1_9 {
public static void main(String[] args) {
Scanner reader = new Scanner(System.in);
System.out.print("Enter 1st number = ");
int num1 = reader.nextInt();
System.out.print("Enter 2nd number = ");
int num2 = reader.nextInt();
System.out.print("Enter 3rd number = ");
int num3 = reader.nextInt();
if(num1>num2 && num1>num3){
System.out.println(num1 + " is Greater from "+ num2 +" "+ num3);
}else if (num2>num1 && num2>num3){
System.out.println(num2 + " is Greater from "+ num1 +" "+ num3);
}else {
System.out.println(num3 + " is Greater from "+ num1 +" "+ num2);
}
}
}
```

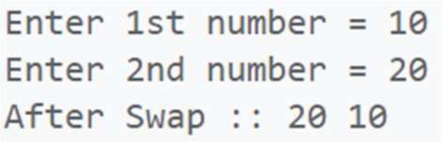


```
Enter 1st number = 30
Enter 2nd number = 50
Enter 3rd number = 60
60 is Greater from 30 50
```

10. Write a Java program to swap two numbers.

Code:

```
import java.util.*;
public class as1_10 {
public static void main(String[] args) {
Scanner reader = new Scanner(System.in);
System.out.print("Enter 1st number = ");
int num1 = reader.nextInt();
System.out.print("Enter 2nd number = ");
int num2 = reader.nextInt();
int temp = num1;
num1 = num2;
num2 = temp;
System.out.println("After Swap :: "+ num1 + " " + num2);
}
}
```

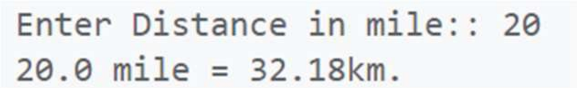


```
Enter 1st number = 10
Enter 2nd number = 20
After Swap :: 20 10
```

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

Code:

```
import java.util.*;
public class as1_11 {
public static void main(String[] args) {
Scanner reader = new Scanner(System.in);
System.out.print("Enter Distance in mile:: ");
double dist = reader.nextDouble();
double kilo = dist*1.609;
System.out.print(dist + " mile = " + kilo +"km.");
}
}
```

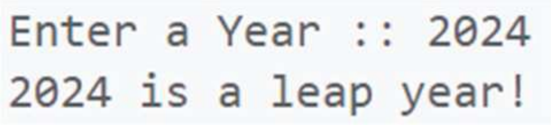


```
Enter Distance in mile:: 20
20.0 mile = 32.18km.
```

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

Code: import java.util.*;

```
public class as1_12 {
public static void main(String[] args) {
Scanner reader = new Scanner(System.in);
System.out.print("Enter a Year :: ");
int year = reader.nextInt();
boolean leap = false;
if (year % 4 == 0) {
if (year % 100 == 0) {
leap = (year % 400 == 0);
} else {
leap = true;
}
}
if (leap) {
System.out.println(year + " is a leap year!");
} else {
System.out.println(year + " is not a leap year!");
}
reader.close();
}
}
```

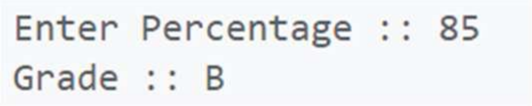


```
Enter a Year :: 2024
2024 is a leap year!
```

13. 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

Code: import java.util.*;

```
public class as1_13 {
public static void main(String[] args) {
Scanner reader = new Scanner(System.in);
System.out.print("Enter Percentage :: ");
double percent = reader.nextDouble();
char grade;
if (percent>=90) {
grade = 'A';
}else if (percent>=80) {
grade = 'B';
}else if (percent>=70) {
grade = 'C';
}else if (percent>=60) {
grade = 'D';
}else if (percent>=40) {
grade = 'E';
}else{
grade = 'F';
}
System.out.println("Grade :: " + grade);
}
}
```

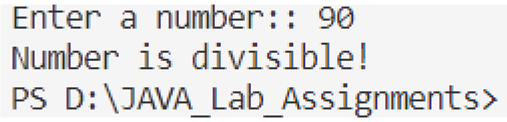


```
Enter Percentage :: 85
Grade :: B
```

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

Code:

```
import java.util.*;
public class as1_14 {
public static void main(String[] args) {
Scanner reader = new Scanner(System.in);
System.out.print("Enter a number:: ");
int num = reader.nextInt();
if (num%5 == 0) {
System.out.println("Number is divisible!");
}else{
System.out.println("Number is not divisible!");
}
}
}
```



```
Enter a number:: 90
Number is divisible!
PS D:\JAVA_Lab_Assignments>
```

NAME: SAGNIK DAS **ROLL NO:** 55

ENROLLMENT NO: 12023006015037 **SEC:** A **Semester:** 2nd

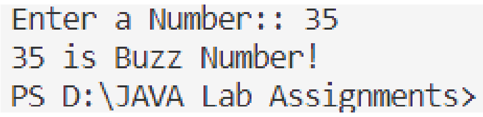
DEPARTMENT: M.C.A.

Week 2

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

Code:

```
import java.util.*;
public class as2_1 {
public static void main(String[] args) {
Scanner reader =  new Scanner(System.in);
System.out.print("Enter a Number:: ");
int num =  reader.nextInt();
boolean buzz = (num%7 == 0 || num%10 == 7);
if (buzz){
System.out.println(num+ " is Buzz Number!");
}else{
System.out.println(num+ " is not a Buzz Number!");
}
}
}
```

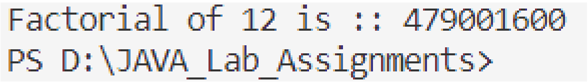


```
Enter a Number:: 35
35 is Buzz Number!
PS D:\JAVA Lab Assignments>
```

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

Code:

```
public class as2_2 {
public static void main(String[] args) {
long fact = factorial(12);
System.out.println("Factorial of 12 is :: " + fact);
}
public static long factorial (int n){
if (n==0||n==1) {
return 1;
}else{
long result = 1;
for(int i = 2; i<=n; i++){
result =  result*i;
}
return result;
}
}
}
```

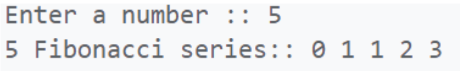


```
Factorial of 12 is :: 479001600
PS D:\JAVA_Lab_Assignments>
```

3. Write a Java program for Fibonacci series.

Code: import java.util.*;

```
public class as2_3 {
public static void main(String[] args) {
Scanner reader =  new Scanner(System.in);
System.out.print("Enter a number :: ");
int num =  reader.nextInt();
System.out.print(num + " Fibonacci series:: ");
fibo(num);
}
public static void fibo (int n){
int first  = 0, second = 1;
for (int i = 1; i<=n; i++){
System.out.print(first + " ");
int temp = first+second;
first = second;
second = temp;
}
}
}
```

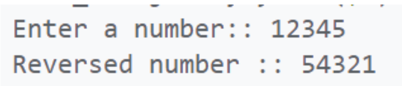


```
Enter a number :: 5
5 Fibonacci series:: 0 1 1 2 3
```

4. Write a Java program to reverse a number.

Code: import java.util.*;

```
public class as2_4 {
public static void main(String[] args) {
Scanner reader = new Scanner(System.in);
System.out.print("Enter a number:: ");
int num = reader.nextInt();
int reversenum = reverse(num);
System.out.println("Reversed number :: "+reversenum);
}
public static int reverse(int n){
int reversenum = 0;
while (n!=0) {
int digit = n%10;
reversenum = reversenum*10+digit;
n= n/10;
}
return reversenum;
}
}
```

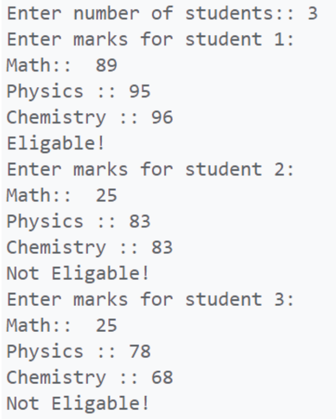


```
Enter a number:: 12345
Reversed number :: 54321
```

5. 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.

Code:

```
import java.util.*;;
public class as2_5 {
public static void main(String[] args) {
Scanner reader = new Scanner(System.in);
System.out.print("Enter number of students:: ");
int num = reader.nextInt();
for(int i = 1; i <= num ; i++){
System.out.println("Enter marks for student " + i + ":");
System.out.print("Math::  ");
int num1 = reader.nextInt();
System.out.print("Physics :: ");
int num2 = reader.nextInt();
System.out.print("Chemistry :: ");
int num3 = reader.nextInt();
int total = num1+num2;
if ((num1>=60 && num2>= 50 && num3>=40)&&( total>= 150)){
System.out.println("Eligable!");
}else{
System.out.println("Not Eligable!");
}
}
}
}
```

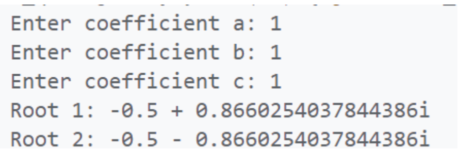


```
Enter number of students:: 3
Enter marks for student 1:
Math::  89
Physics :: 95
Chemistry :: 96
Eligable!
Enter marks for student 2:
Math::  25
Physics :: 83
Chemistry :: 83
Not Eligable!
Enter marks for student 3:
Math::  25
Physics :: 78
Chemistry :: 68
Not Eligable!
```

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

Code:

```
import java.util.*;
public class as2_6 {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter coefficient a: ");
double a = scanner.nextDouble();
System.out.print("Enter coefficient b: ");
double b = scanner.nextDouble();
System.out.print("Enter coefficient c: ");
double c = scanner.nextDouble();
double discriminant = b * b - 4 * a * c;
if (discriminant > 0) {
double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
System.out.println("Root 1: " + root1);
System.out.println("Root 2: " + root2);
} else if (discriminant == 0) {
double root = -b / (2 * a);
System.out.println("Roots are real and equal: " + root);
} else {
double realPart = -b / (2 * a);
double imaginaryPart = Math.sqrt(-discriminant) / (2 * a);
System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
}
scanner.close();
}
}
```

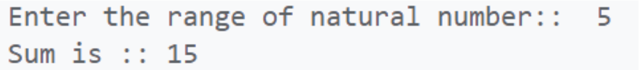


```
Enter coefficient a: 1
Enter coefficient b: 1
Enter coefficient c: 1
Root 1: -0.5 + 0.8660254037844386i
Root 2: -0.5 - 0.8660254037844386i
```

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

Code:

```
import java.util.*;
public class sum_natural {
public static void main(String[] args) {
Scanner reader =  new Scanner(System.in);
System.out.print("Enter the range of natural number::  ");
int num = reader.nextInt();
int sum = total(num);
System.out.println("Sum is :: "+sum);
}
public static int total(int num){
int sum = 0;
for(int i = 0; i<=num ; i++){
sum= sum+i;
}
return sum;
}
}
```

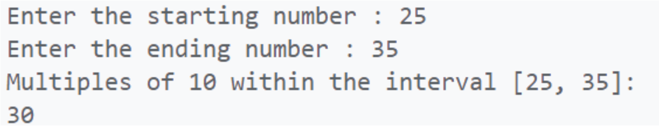


```
Enter the range of natural number::  5
Sum is :: 15
```

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

Code:

```
import java.util.*;
public class as2_8 {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the starting number : ");
int start = scanner.nextInt();
System.out.print("Enter the ending number : ");
int end = scanner.nextInt();
System.out.println("Multiples of 10 within the interval [" + start + ", " + end + "]:");
mul(start, end);
scanner.close();
}
public static void mul(int start, int end) {
if (start % 10 != 0) {
start += 10 - (start % 10);
}
for (int i = start; i <= end; i += 10) {
System.out.println(i);
}
}
}
```

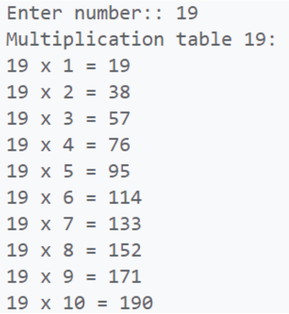


```
Enter the starting number : 25
Enter the ending number : 35
Multiples of 10 within the interval [25, 35]:
30
```

9. Write a Java program to generate multiplication table.

Code: import java.util.*;

```
public class as2_9 {
public static void main(String[] args) {
Scanner reader  = new Scanner(System.in);
System.out.print("Enter number:: ");
int num =  reader.nextInt();
table(num);
}
public static void table(int n){
System.out.println("Multiplication table " + n + ":");
for (int i = 1; i <= 10; i++) {
System.out.println(n + " x " + i + " = " + (n * i));
}
}
}
```

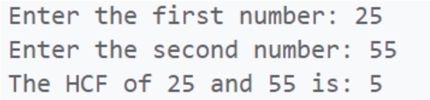


```
Enter number:: 19
Multiplication table 19:
19 x 1 = 19
19 x 2 = 38
19 x 3 = 57
19 x 4 = 76
19 x 5 = 95
19 x 6 = 114
19 x 7 = 133
19 x 8 = 152
19 x 9 = 171
19 x 10 = 190
```

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

Code:

```
import java.util.*;
public class HCFFinder {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the first number: ");
int num1 = scanner.nextInt();
System.out.print("Enter the second number: ");
int num2 = scanner.nextInt();
int hcf = findHCF(num1, num2);
System.out.println("The HCF of " + num1 + " and " + num2 + " is: " + hcf);
scanner.close();
}
public static int findHCF(int num1, int num2) {
int small = Math.min(num1, num2);
int hcf = 1;
for (int i = 1; i <= small; i++) {
if (num1 % i == 0 && num2 % i == 0) {
hcf = i;
}
}
return hcf;
}
}
```

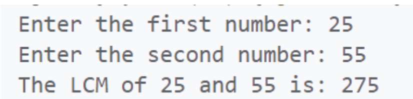


```
Enter the first number: 25
Enter the second number: 55
The HCF of 25 and 55 is: 5
```

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

Code: import java.util.*;

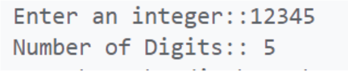
```
public class LCM {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the first number: ");
int num1 = scanner.nextInt();
System.out.print("Enter the second number: ");
int num2 = scanner.nextInt();
int lcm = findLCM(num1, num2);
System.out.println("The LCM of " + num1 + " and " + num2 + " is: " + lcm);
scanner.close();
}
public static int findLCM(int num1, int num2) {
int max = Math.max(num1, num2);
while (true) {
if (max % num1 == 0 && max % num2 == 0) {
return max;
}
max++;
}
}
}
```



```
Enter the first number: 25
Enter the second number: 55
The LCM of 25 and 55 is: 275
```

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

Code:
```
import java.util.*;
public class count_digit {
public static void main(String[] args){
Scanner read = new Scanner(System.in);
System.out.print("Enter an integer::");
int num = read.nextInt();
int count = 0;
int temp  = num;
while (temp!=0) {
temp = temp/10;
count ++;
}
System.out.println("Number of Digits:: " +count);
}
}
```

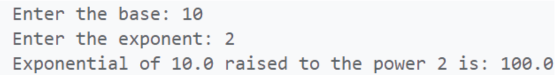


```
Enter an integer::12345
Number of Digits:: 5
```

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

Code:
```
import java.util.*;
public class exponent {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the base: ");
double base = scanner.nextDouble();
System.out.print("Enter the exponent: ");
int exponent = scanner.nextInt();
double result = calculate(base, exponent);
System.out.println("Exponential of " + base + " raised to the power " + exponent + " is: " + result);
}
public static double calculate(double base, int exponent) {
```

```
double result = 1;
for (int i = 0; i < Math.abs(exponent); i++) {
result *= base;
}
if (exponent < 0) {
result = 1 / result;
}
return result;
}
}
```

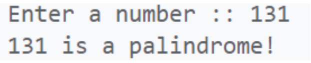


```
Enter the base: 10
Enter the exponent: 2
Exponential of 10.0 raised to the power 2 is: 100.0
```

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

Code: import java.util.*;

```
public class palindrome {
public static void main(String[] args) {
Scanner read  = new Scanner(System.in);
System.out.print("Enter a number :: ");
int num = read.nextInt();
boolean palin = check(num);
if (palin) {
System.out.println(num+ " is a palindrome!");
}else{
System.out.println(num+ " is not a palindrome!");
}
}
public static boolean check (int n){
int temp = n;
int rev = 0;
while (n!=0) {
int digit = n%10;
rev = rev*10+digit;
n = n/10;
}
return temp==rev;
}
}
```

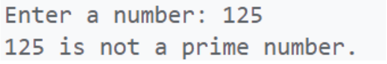


```
Enter a number :: 131
131 is a palindrome!
```

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

Code: import java.util.*;

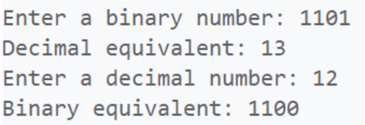
```
public class prime {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter a number: ");
int num = scanner.nextInt();
boolean isPrime = checkPrime(num);
if (isPrime) {
System.out.println(num + " is a prime number.");
} else {
System.out.println(num + " is not a prime number.");
}
scanner.close();
}
public static boolean checkPrime(int num) {
if (num <= 1) {
return false;
}
for (int i = 2; i <= Math.sqrt(num); i++) {
if (num % i == 0) {
return false;
}
}

return true;

}

}
```



```
Enter a number: 125
125 is not a prime number.
```

16. Write a Java program to convert a Binary Number to Decimal and Decimal to Binary. Code: import java.util.*;

```
public class binary_dec {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a binary number: ");

String binary = scanner.nextLine();

int decimal = binary_decimal(binary);

System.out.println("Decimal equivalent: " + decimal);

System.out.print("Enter a decimal number: ");

int number = scanner.nextInt();

String binaryEquivalent = decimal_binary(number);

System.out.println("Binary equivalent: " + binaryEquivalent);

scanner.close();

}

public static int binary_decimal(String binary) {

int decimal = 0;

int power = 0;

for (int i = binary.length() - 1; i >= 0; i--) {

if (binary.charAt(i) == '1') {

decimal += Math.pow(2, power);

}

power++;

}

return decimal;
}
public static String decimal_binary(int number) {
StringBuilder binary = new StringBuilder();
while (number > 0) {
binary.insert(0, number % 2);
number /= 2;
}
return binary.toString();
}
}
```



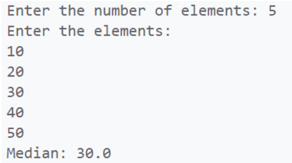
```
Enter a binary number: 1101
Decimal equivalent: 13
Enter a decimal number: 12
Binary equivalent: 1100
```

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

Code: import java.util.Arrays;

```
import java.util.*;
public class median {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the number of elements: ");
int n = scanner.nextInt();
double[] numbers = new double[n];
System.out.println("Enter the elements:");
for (int i = 0; i < n; i++) {
numbers[i] = scanner.nextDouble();
}
Arrays.sort(numbers);
double median;
if (n % 2 == 0) {
median = (numbers[n / 2 - 1] + numbers[n / 2]) / 2;
} else {
```

```
median = numbers[n / 2];
}
System.out.println("Median: " + median);
}
}
```

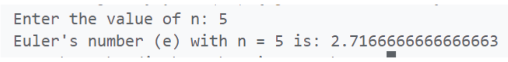


```
Enter the number of elements: 5
Enter the elements:
10
20
30
40
50
Median: 30.0
```

18. 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!

Code:

```
import java.util.Scanner;
public class EulerNumber {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the value of n: ");
int n = scanner.nextInt();
double e = calculateEulerNumber(n);
System.out.println("Euler's number (e) with n = " + n + " is: " + e);
scanner.close();
}
public static double calculateEulerNumber(int n) {
double e = 1;
double factorial = 1;
for (int i = 1; i <= n; i++) {
factorial *= i;
e += 1.0 / factorial;
}
return e;
}
}
```

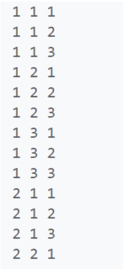


```
Enter the value of n: 5
Euler's number (e) with n = 5 is: 2.7166666666666663
```

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

Code: public class combinations_123 {

```
public static void main(String[] args) {
generateCombinations();
}
public static void generateCombinations() {
for (int i = 1; i <= 3; i++) {
for (int j = 1; j <= 3; j++) {
for (int k = 1; k <= 3; k++) {
System.out.println(i + " " + j + " " + k);
}
}
}
}
}
```



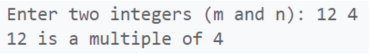
```
1 1 1
1 1 2
1 1 3
1 2 1
1 2 2
1 2 3
1 3 1
1 3 2
1 3 3
2 1 1
2 1 2
2 1 3
2 2 1
```

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

Code: import java.util.Scanner;

```
public class multiple_checker {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter two integers (m and n): ");
int m = scanner.nextInt();
int n = scanner.nextInt();
if (m % n == 0) {
```

```
System.out.println(m + " is a multiple of " + n);
} else {
System.out.println(m + " is not a multiple of " + n);
}
scanner.close();
}
}
```

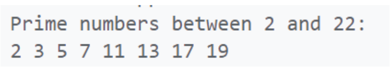


```
Enter two integers (m and n): 12 4
12 is a multiple of 4
```

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

Code: import java.util.Scanner;

```
public class prime_interval {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the lower bound of the interval: ");
int lowerBound = scanner.nextInt();
System.out.print("Enter the upper bound of the interval: ");
int upperBound = scanner.nextInt();
System.out.println("Prime numbers between " + lowerBound + " and " + upperBound
+ ":");
for (int i = lowerBound; i <= upperBound; i++) {
if (isPrime(i)) {
System.out.print(i + " ");
}
}
scanner.close();
}
public static boolean isPrime(int number) {
if (number <= 1) {
return false;
}
for (int i = 2; i <= Math.sqrt(number); i++) {
if (number % i == 0) {
return false;
}
}
return true;
}
}
```

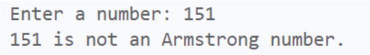


```
Prime numbers between 2 and 22:
2 3 5 7 11 13 17 19
```

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

Code:

```
import java.util.Scanner;
public class armstrong_num {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter a number: ");
int number = scanner.nextInt();
if (isArmstrongNumber(number)) {
System.out.println(number + " is an Armstrong number.");
} else {
System.out.println(number + " is not an Armstrong number.");
}
scanner.close();
}
public static boolean isArmstrongNumber(int number) {
int originalNumber = number;
int sum = 0;
int digits = String.valueOf(number).length();
while (number > 0) {
int digit = number % 10;
sum += Math.pow(digit, digits);

number /= 10;

}

return originalNumber == sum;

}

}
```

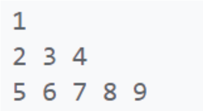


```
Enter a number: 151
151 is not an Armstrong number.
```

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

```
23. 1
    2 3 4
    5 6 7 8 9
```

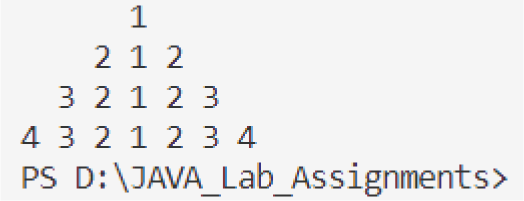
Code:
```
public class patternprinter_01 {

public static void main(String[] args) {

int num = 1;

int count = 1;

for (int i = 1; i <= 3; i++) {

for (int j = 1; j <= count; j++) {

System.out.print(num + " ");

num++;

}

System.out.println();

count += 2;

}

}

}
```



```
1
2 3 4
5 6 7 8 9
```

```
24.       1
        2 1 2
      3 2 1 2 3
    4 3 2 1 2 3 4
```

Code:
```
public class patternprinter_02 {
public static void main(String[] args) {
int rows = 4;
for (int i = 1; i <= rows; i++) {
for (int j = rows - i; j >= 1; j--) {
System.out.print("  ");
}
for (int k = i; k >= 1; k--) {
System.out.print(k + " ");
}
for (int l = 2; l <= i; l++) {
System.out.print(l + " ");
}
System.out.println();
}
}
}
```

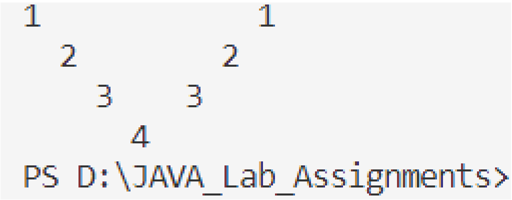


```
      1
    2 1 2
  3 2 1 2 3
4 3 2 1 2 3 4
PS D:\JAVA_Lab_Assignments>
```

25.
```
1     1
 2   2
  3 3
   4
```

Code:
```
public class patternprinter_03 {
public static void main(String[] args) {
int rows = 4;
for (int i = 1; i <= rows; i++) {
for (int j = 1; j < i; j++) {
```

```
System.out.print("  ");
}
System.out.print(i);
for (int k = 1; k <= 4 * (rows - i); k++) {
System.out.print(" ");
}
if (i < rows) {
System.out.println(i);
} else {
System.out.println();
}
}
}
}
```



```
1            1
  2        2
    3    3
      4
PS D:\JAVA_Lab_Assignments>
```

NAME: SAGNIK DAS **ROLL NO:** 55

ENROLLMENT NO: 12023006015037 **SEC:** A **Semester:** 2nd

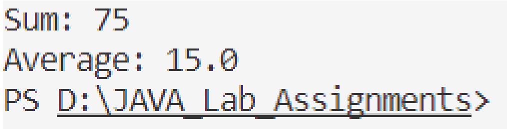
DEPARTMENT: M.C.A.

Week 3

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

Code:

```
public class SumAndAverage {
public static void main(String[] args) { int[] array = {5, 10, 15, 20, 25};
int sum = 0;
for (int num : array) {
sum += num;
}
double average = (double) sum / array.length;
System.out.println("Sum: " + sum); System.out.println("Average: " + average); } }
```

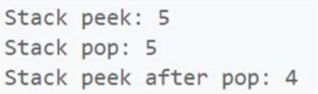


```
Sum: 75
Average: 15.0
PS D:\JAVA_Lab_Assignments>
```

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

Code:

```
public class Stack {
private int maxSize;
private int[] stackArray;
 private int top;
public Stack(int size)
{ maxSize = size;
stackArray = new int[maxSize]; top = -1;
}
public void push(int value)
{ if (isFull()) {
System.out.println("Stack is full. Cannot push element."); return;
}
top++;
stackArray[top] = value;
}
public int pop()
{ if (isEmpty()) {
System.out.println("Stack is empty. Cannot pop element."); return -1;
}
int poppedValue = stackArray[top]; top--;
return poppedValue;
}
public int peek()
{ if (isEmpty()) {
System.out.println("Stack is empty. Cannot peek element."); return -1;
}
return stackArray[top];
}
public boolean isEmpty() { return top == -1;
}
public boolean isFull() { return top == maxSize - 1;
}
public static void main(String[] args)
{ Stack stack = new Stack(5);
stack.push(1);
stack.push(2);
stack.push(3);
stack.push(4); stack.push(5);
System.out.println("Stack peek: " + stack.peek()); System.out.println("Stack pop: " + stack.pop()); System.out.println("Stack peek after pop: " + stack.peek());
}
}
```



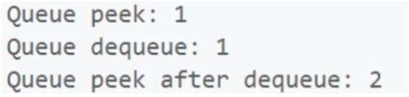
```
Stack peek: 5
Stack pop: 5
Stack peek after pop: 4
```

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

Code: public class Queue {

```
private int maxSize;
private int[] queueArray;
private int front;
private int rear; private int size;
public Queue(int size)
{ maxSize = size;
queueArray = new int[maxSize];
front = 0;
rear = -1;
size = 0;
}
```

```
public void enqueue(int value)
{ if (isFull()) {
System.out.println("Queue is full. Cannot enqueue element.");
return;
}
rear = (rear + 1) % maxSize;
queueArray[rear] = value; size++;
}
public int dequeue() { if (isEmpty()) {
System.out.println("Queue is empty. Cannot dequeue element.");
return -1;
}
int dequeuedValue = queueArray[front];
front = (front + 1) % maxSize;
size--;
return dequeuedValue;
}
public int peek() { if (isEmpty()) {
System.out.println("Queue is empty. Cannot peek element."); return -1;
}
return queueArray[front];
}
public boolean isEmpty() {
return size == 0;
}
public boolean isFull() { return size == maxSize;
}
public static void main(String[] args)
{ Queue queue = new Queue(5);
queue.enqueue(1);
queue.enqueue(2);
queue.enqueue(3);
queue.enqueue(4);
queue.enqueue(5);
System.out.println("Queue peek: " + queue.peek());
System.out.println("Queue dequeue: " + queue.dequeue());
System.out.println("Queue peek after dequeue: " + queue.peek());
}
}
```

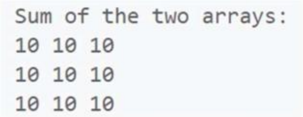


```
Queue peek: 1
Queue dequeue: 1
Queue peek after dequeue: 2
```

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

Code: public class SumOfArrays {

```
public static void main(String[] args) {
int[][] array1 = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
int[][] array2 = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
int rows = array1.length;
int cols = array1[0].length;
int[][] sumArray = new int[rows][cols];
for (int i = 0; i < rows; i++) {
for (int j = 0; j < cols; j++) {
sumArray[i][j] = array1[i][j] + array2[i][j];
}
}
System.out.println("Sum of the two arrays:");
for (int i = 0; i < rows; i++) {
for (int j = 0; j < cols; j++)
{ System.out.print(sumArray[i][j] + " ");
}
System.out.println();
}
}
}
```

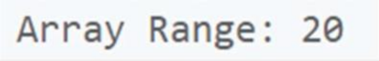


```
Sum of the two arrays:
10 10 10
10 10 10
10 10 10
```

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

Code:

```
public class ArrayRange {
public static void main(String[] args)
{ int[] array = {5, 10, 15, 20, 25};
int min = array[0];
int max = array[0];
for (int i = 1; i < array.length; i++)
{ if (array[i] < min) {
min = array[i];
}
if (array[i] > max) { max = array[i];
}
}
int range = max - min;
System.out.println("Array Range: " + range);
}
}
```

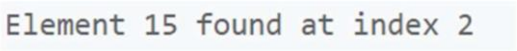


```
Array Range: 20
```

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

Code:

```
public class ElementSearch {
public static void main(String[] args)
{ int[] array = {5, 10, 15, 20, 25};
int target = 15;
boolean found = false;
for (int i = 0; i < array.length; i++)
{ if (array[i] == target) {
found = true;
System.out.println("Element " + target + " found at index " + i); break;
}
}
if (!found) {
System.out.println("Element " + target + " not found in the array.");
}
}
}
```

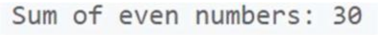


```
Element 15 found at index 2
```

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

Code:

```
public class SumOfEvenNumbers {
public static void main(String[] args) {
int[] array = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int sumOfEvens = 0; for (int num : array) { if (num % 2 == 0) {
sumOfEvens += num;
}
}
System.out.println("Sum of even numbers: " + sumOfEvens);
}
}
```

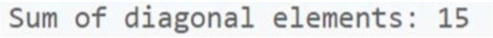


```
Sum of even numbers: 30
```

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

Code:

```
public class DiagonalSum {
public static void main(String[] args) {
int[][] array = {
{1, 2, 3},
{4, 5, 6},
{7, 8, 9}
};
int sum = 0;
for (int i = 0; i < array.length; i++)
{ sum += array[i][i];
}
System.out.println("Sum of diagonal elements: " + sum);
}
}
```

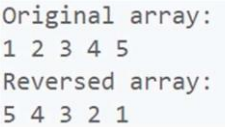


```
Sum of diagonal elements: 15
```

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

Code:

```
public class ReverseArray {
public static void main(String[] args) {
int[] array = {1, 2, 3, 4, 5};
System.out.println("Original array:");
printArray(array);
int length = array.length;
for (int i = 0; i < length / 2; i++) {
int temp = array[i];
array[i] = array[length - 1 - i];
array[length - 1 - i] = temp;
}
System.out.println("Reversed array:");
printArray(array);
}
public static void printArray(int[] arr) {
for (int num : arr) {
System.out.print(num + " ");
}
System.out.println();
}
}
```

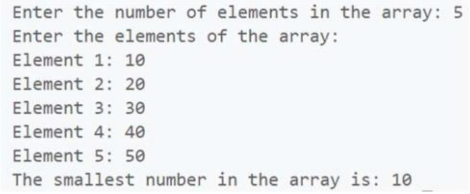


```
Original array:
1 2 3 4 5
Reversed array:
5 4 3 2 1
```

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

Code:

```
import java.util.Scanner;
public class SmallestNumber {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the number of elements in the array: ");
int n = scanner.nextInt();
int[] array = new int[n];
System.out.println("Enter the elements of the array:");
for (int i = 0; i < n; i++) {
System.out.print("Element " + (i + 1) + ": ");
array[i] = scanner.nextInt();
int smallest = array[0]; for (int i = 1; i < n; i++) {
if (array[i] < smallest) { smallest = array[i];
}
}
System.out.println("The smallest number in the array is: " + smallest);
scanner.close();
}
}
```

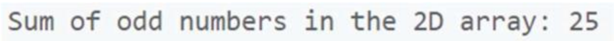


```
Enter the number of elements in the array: 5
Enter the elements of the array:
Element 1: 10
Element 2: 20
Element 3: 30
Element 4: 40
Element 5: 50
The smallest number in the array is: 10 _
```

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

Code:

```
public class SumOfOddNumbers {
public static void main(String[] args) { int[][] array = {
{1, 2, 3},
{4, 5, 6},
{7, 8, 9}
};
int sum = 0;
for (int[] row : array)
{ for (int num : row) {
if (num % 2 != 0)
{ sum += num;
}
}
}
System.out.println("Sum of odd numbers in the 2D array: " + sum);
}
}
```



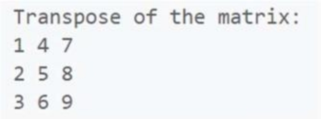
```
Sum of odd numbers in the 2D array: 25
```

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

Code: public class MatrixTranspose {

```
public static void main(String[] args) { int[][] matrix = {
{1, 2, 3},
{4, 5, 6},
{7, 8, 9}
};
int rows = matrix.length; int cols = matrix[0].length;
int[][] transpose = new int[cols][rows];
for (int i = 0; i < cols; i++) {
for (int j = 0; j < rows; j++)
{ transpose[i][j] = matrix[j][i];
}
System.out.println("Transpose of the matrix:");
for (int i = 0; i < cols; i++) {
```

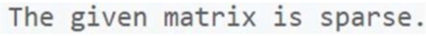
```
for (int j = 0; j < rows; j++)
{ System.out.print(transpose[i][j] + " ");
}
System.out.println();
}
}
}
```



```
Transpose of the matrix:
1 4 7
2 5 8
3 6 9
```

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

Code:
```
public class SparseMatrix {
public static void main(String[] args) {
 int[][] matrix = {
{1, 0, 0},
{0, 0, 0},
{0, 0, 3}
};
int zeroCount = 0;
int nonZeroCount = 0;
for (int[] row : matrix) {
for (int element : row)
{ if (element == 0)
{ zeroCount++;
} else {
nonZeroCount++;
}
}
}
if (zeroCount > nonZeroCount)
{ System.out.println("The given matrix is sparse.");
} else {
System.out.println("The given matrix is not sparse.");
}
}
}
```

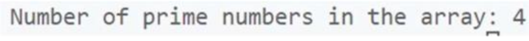


```
The given matrix is sparse.
```

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

Code:

```
public class PrimeNumberCount {
public static void main(String[] args)
{ int[] array = {2, 3, 4, 5, 6, 7, 8, 9, 10};
int primeCount = 0;
for (int num : array) {
if (isPrime(num))
{ primeCount++;
}
}
System.out.println("Number of prime numbers in the array: " + primeCount);
public static boolean isPrime(int num)
{ if (num <= 1) {
return false;
}
for (int i = 2; i <= Math.sqrt(num); i++)
{ if (num % i == 0) {
return false;
}
}
return true;
}
}
```

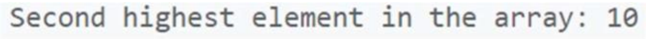


```
Number of prime numbers in the array: 4
```

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

Code:

```
public class SecondHighestElement {
public static void main(String[] args)
{ int[] array = {5, 10, 3, 8, 15, 7};
int max = Integer.MIN_VALUE;
int secondMax = Integer.MIN_VALUE;
for (int num : array) {
if (num > max)
{ secondMax = max; max = num;
}
else if (num > secondMax && num != max)
{ secondMax = num;
}
}
if (secondMax != Integer.MIN_VALUE) {
System.out.println("Second highest element in the array: " + secondMax);
} else {
System.out.println("Second highest element does not exist in the array.");
}
}
}
```

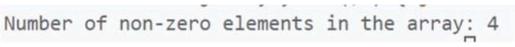


```
Second highest element in the array: 10
```

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

Code:

```
public class NonZeroElementCount {
public static void main(String[] args)
{ int[] array = {0, 5, 0, 10, 0, 15, 20};
int nonZeroCount = 0;
for (int num : array) {
if (num != 0)
{ nonZeroCount++;
}
}
System.out.println("Number of non-zero elements in the array: " + nonZeroCount);
}
}
```

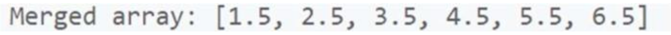


```
Number of non-zero elements in the array: 4
```

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

Code:

```
import java.util.Arrays;
public class MergeFloatArrays {
public static void main(String[] args)
{ float[] array1 = {1.5f, 2.5f, 3.5f};
float[] array2 = {4.5f, 5.5f, 6.5f};
int mergedLength = array1.length + array2.length;
float[] mergedArray = new float[mergedLength];
System.arraycopy(array1, 0, mergedArray, 0, array1.length);
System.arraycopy(array2, 0, mergedArray, array1.length, array2.length);
System.out.println("Merged array: " + Arrays.toString(mergedArray));
}
}
```

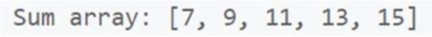


```
Merged array: [1.5, 2.5, 3.5, 4.5, 5.5, 6.5]
```

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

Code:

```
import java.util.Arrays;
public class AddArrays {
public static void main(String[] args)
{ int[] array1 = {1, 2, 3, 4, 5};
int[] array2 = {6, 7, 8, 9, 10};
int[] sumArray = new int[array1.length];
for (int i = 0; i < array1.length; i++) {
sumArray[i] = array1[i] + array2[i];
}
System.out.println("Sum array: " + Arrays.toString(sumArray));
}
}
```

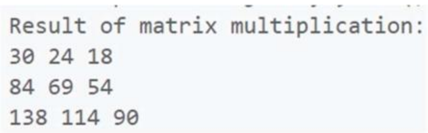


```
Sum array: [7, 9, 11, 13, 15]
```

19. Write a Java program to multiply two matrices.

Code:

```
public class MatrixMultiplication {
public static void main(String[] args)
{ int[][] matrix1 = {
{1, 2, 3},
{4, 5, 6},
{7, 8, 9}
};
int[][] matrix2 = {
{9, 8, 7},
{6, 5, 4},
{3, 2, 1}
};
int rows1 = matrix1.length;
int cols1 = matrix1[0].length;
int cols2 = matrix2[0].length;
int[][] result = new int[rows1][cols2];
for (int i = 0; i < rows1; i++) {
for (int j = 0; j < cols2; j++) {
for (int k = 0; k < cols1; k++) {
result[i][j] += matrix1[i][k] * matrix2[k][j];
}
}
}
System.out.println("Result of matrix multiplication:");
printMatrix(result);
}
public static void printMatrix(int[][] matrix)
{ for (int[] row : matrix)
{
for (int num : row) { System.out.print(num + " ");
}
System.out.println();
}
}
}
```

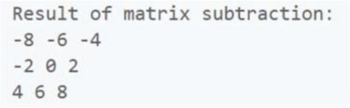


```
Result of matrix multiplication:
30 24 18
84 69 54
138 114 90
```

20. Write a Java program to subtract two matrices. Code:

```
public class MatrixSubtraction {
public static void main(String[] args)
{ int[][] matrix1 = {
{1, 2, 3},
{4, 5, 6},
{7, 8, 9}
};
int[][] matrix2 = {
{9, 8, 7},
{6, 5, 4},
{3, 2, 1}
};
int rows = matrix1.length;
int cols = matrix1[0].length;
int[][] result = new int[rows][cols];
for (int i = 0; i < rows; i++) {
```

```
for (int j = 0; j < cols; j++) {
result[i][j] = matrix1[i][j] - matrix2[i][j];
}
}
System.out.println("Result of matrix subtraction:");
printMatrix(result);
}
public static void printMatrix(int[][] matrix)
{ for (int[] row : matrix) {
for (int num : row)
{ System.out.print(num + " ");
}
System.out.println();
}
}
}
```

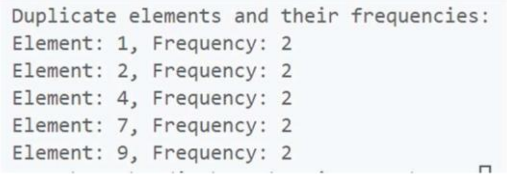


```
Result of matrix subtraction:
-8 -6 -4
-2 0 2
4 6 8
```

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

Code:

```
import java.util.HashMap;
import java.util.Map;
public class DuplicateElements {
public static void main(String[] args) {
int[] array = {1, 2, 3, 2, 4, 5, 1, 6, 4, 7, 8, 7, 9, 9};
Map<Integer, Integer> frequencyMap = new HashMap<>();
for (int num : array) {
frequencyMap.put(num, frequencyMap.getOrDefault(num, 0) + 1);
}
System.out.println("Duplicate elements and their frequencies:");
for (Map.Entry<Integer, Integer> entry : frequencyMap.entrySet())
{ if (entry.getValue() > 1) {
System.out.println("Element: " + entry.getKey() + ", Frequency: " + entry.getValue());
}
}
}
}
```

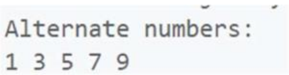


```
Duplicate elements and their frequencies:
Element: 1, Frequency: 2
Element: 2, Frequency: 2
Element: 4, Frequency: 2
Element: 7, Frequency: 2
Element: 9, Frequency: 2
```

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

Code:

```
public class AlternateNumbers {
public static void printAlternateNumbers(int[] array)
{ for (int i = 0; i < array.length; i += 2) {
System.out.print(array[i] + " ");
}
System.out.println();
}
public static void main(String[] args) {
int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
System.out.println("Alternate numbers:");
printAlternateNumbers(numbers);
}
}
```

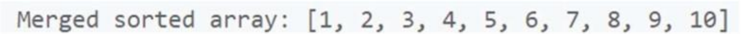


```
Alternate numbers:
1 3 5 7 9
```

23. 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.

Code:

```
import java.util.Arrays;
public class MergeSortedArrays {
public static void main(String[] args)
{ int[] arrayA = {1, 3, 5, 7, 9};
int[] arrayB = {2, 4, 6, 8, 10};


int lengthA = arrayA.length; int lengthB = arrayB.length;
int[] mergedArray = new int[lengthA + lengthB]; int indexA = 0;
int indexB = 0;
int indexC = 0;
while (indexA < lengthA && indexB < lengthB)
{ if (arrayA[indexA] < arrayB[indexB]) {
mergedArray[indexC++] = arrayA[indexA++];
} else {
mergedArray[indexC++] = arrayB[indexB++];
}
}
while (indexA < lengthA) { mergedArray[indexC++] = arrayA[indexA++];
}
while (indexB < lengthB) { mergedArray[indexC++] = arrayB[indexB++];
}
System.out.println("Merged sorted array: " + Arrays.toString(mergedArray));
}
}
```

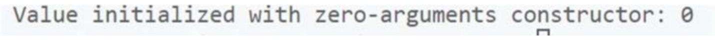


```
Merged sorted array: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

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

Code:

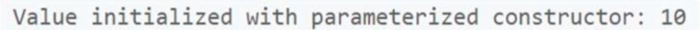
```
public class ZeroArgumentsConstructor {
private int value;
public ZeroArgumentsConstructor()
{ this.value = 0;
}
public int getValue()
{ return this.value;
}
public static void main(String[] args) {
ZeroArgumentsConstructor obj = new ZeroArgumentsConstructor();
System.out.println("Value initialized with zero-arguments constructor: " +
obj.getValue());
}
}
```



```
Value initialized with zero-arguments constructor: 0
```

25. Write a Java program to show parameterized constructor.

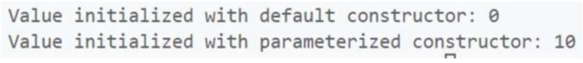
Code:
```
public class ParameterizedConstructor {
private int value;
public ParameterizedConstructor(int value)
{ this.value = value;
}
public int getValue()
{ return this.value;
}
public static void main(String[] args) {
ParameterizedConstructor obj = new ParameterizedConstructor(10);
System.out.println("Value initialized with parameterized constructor: " +
obj.getValue());
}
}
```



```
Value initialized with parameterized constructor: 10
```

26. Write a Java program to show constructor overloading.

Code:
```
public class ConstructorOverloading {
private int value;
public ConstructorOverloading()
{ this.value = 0;
}
public ConstructorOverloading(int value) {
this.value = value;
}
public int getValue() { return this.value;
}
public static void main(String[] args) {
ConstructorOverloading obj1 = new ConstructorOverloading();
ConstructorOverloading obj2 = new ConstructorOverloading(10);
System.out.println("Value initialized with default constructor: " + obj1.getValue());
System.out.println("Value initialized with parameterized constructor: " +
obj2.getValue());
}
}
```



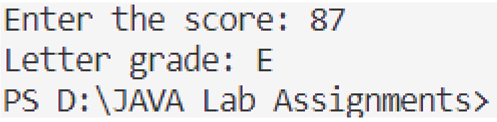
```
Value initialized with default constructor: 0
Value initialized with parameterized constructor: 10
```

27. 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.

Code:

```
import java.util.Scanner; class Grader {
private int score;
public Grader(int score) { this.score = score;
}
public String letterGrade() {
if (score >= 90 && score <= 100) { return "O";
} else if (score >= 80 && score < 90) { return "E";
} else if (score >= 70 && score < 80) { return "A";
} else if (score >= 60 && score < 70) { return "B";
} else if (score >= 50 && score < 60) { return "C";
} else {
return "F";
}
}
}
public class GraderDemo {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the score: ");
int score = scanner.nextInt();
if (score < 0 || score > 100) {
System.out.println("Invalid score! Score must be between 0 and 100.");
} else {
Grader grader = new Grader(score);
String grade = grader.letterGrade();
System.out.println("Letter grade: " + grade);
}
scanner.close();
}
}
```



```
Enter the score: 87
Letter grade: E
PS D:\JAVA Lab Assignments>
```

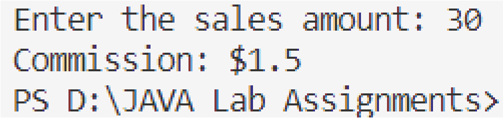
28. Write a class, Commission, which has an instance variable, sales; an appropriate constructor; and a method, commission() that returns the commission. Now write a demo class to test the Commission class by reading a sale from the user, using it to create a Commission object after validating that the 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".

Code: import java.util.Scanner;

```
class Commission {
private double sales;
public Commission(double sales)
{ this.sales = sales;
}
```

```
public double commission()
{ if (sales < 0) {
return -1;
} else if (sales <= 1000) {
return 0.05 * sales;
} else if (sales <= 5000)
{ return 0.08 * sales;
} else
{return 0.1 * sales;
}
}
}
public class CommissionDemo {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the sales amount: ");
double sales = scanner.nextDouble();
if (sales < 0) {
System.out.println("Invalid Input");
} else {
Commission commission = new Commission(sales);
double commissionAmount = commission.commission();
if (commissionAmount == -1) {
System.out.println("Invalid Input");
} else {
System.out.println("Commission: $" + commissionAmount);
}
}
scanner.close();
}
```

```
}
```



```
Enter the sales amount: 30
Commission: $1.5
PS D:\JAVA Lab Assignments>
```

NAME: SAGNIK DAS **ROLL NO:** 55

ENROLLMENT NO: 12023006015037 **SEC:** A **Semester:** 2nd

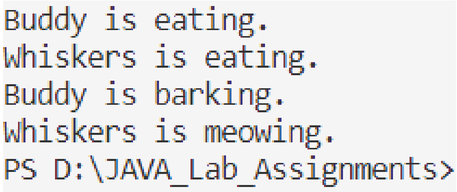
DEPARTMENT: M.C.A.

Week 4

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

Code:

```
class Animal {
String name;
public Animal(String name)
{ this.name = name;
}
public void eat()
{ System.out.println(name + " is eating.");
}
}
class Dog extends Animal
{ public Dog(String name) {
super(name);
}
public void bark()
{ System.out.println(name + " is barking.");
}
}
class Cat extends Animal
{ public Cat(String name) {
super(name);
}
public void meow()
{ System.out.println(name + " is meowing.");
}
}
public class InheritanceExample
{ public static void main(String[] args) {
Dog myDog = new Dog("Buddy");
Cat myCat = new Cat("Whiskers");
myDog.eat();
myCat.eat();
myDog.bark();
myCat.meow();
}
}
```



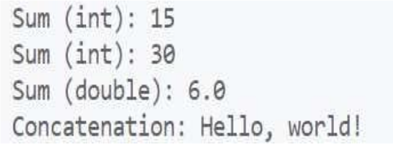
```
Buddy is eating.
Whiskers is eating.
Buddy is barking.
Whiskers is meowing.
PS D:\JAVA_Lab_Assignments>
```

2. Write a Java program to show method overloading.

Code: public class MethodOverloadingExample {

```
public int add(int a, int b)
{ return a + b;
}
public int add(int a, int b, int c)
{ return a + b + c;
}
public double add(double a, double b)
{ return a + b;
}
public String concatenate(String str1, String str2)
{ return str1 + str2;
}
```

```
public static void main(String[] args) {
MethodOverloadingExample example = new MethodOverloadingExample();
System.out.println("Sum (int): " + example.add(5, 10));
System.out.println("Sum (int): " + example.add(5, 10, 15));
System.out.println("Sum (double): " + example.add(3.5, 2.5));
System.out.println("Concatenation: " + example.concatenate("Hello, ", "world!"));
}
}
```

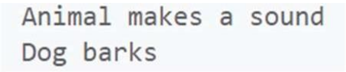


```
Sum (int): 15
Sum (int): 30
Sum (double): 6.0
Concatenation: Hello, world!
```

3. Write a Java program to show method overriding.

Code:

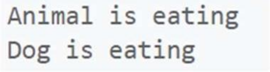
```
class Animal {
public void sound()
{ System.out.println("Animal makes a sound");
}
}
class Dog extends Animal
{ public void sound() {
System.out.println("Dog barks");
}
}
public class MethodOverridingExample
{ public static void main(String[] args) {
Animal animal = new Animal();
Dog dog = new Dog();
animal.sound();
dog.sound();
}
}
```



```
Animal makes a sound
Dog barks
```

4. Write a Java program to show method hiding. Code: class Animal {

```
public static void eat()
{ System.out.println("Animal is eating");
}
}
class Dog extends Animal { public static void eat()
{System.out.println("Dog is eating");
}
}
public class MethodHidingExample
{ public static void main(String[] args) {
Animal.eat();
Dog.eat();
}
}
```



```
Animal is eating
Dog is eating
```

5. 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.

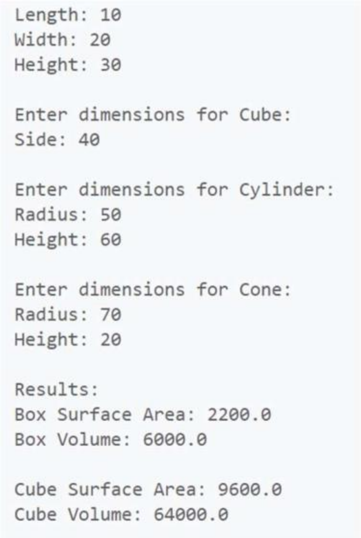
Code: import java.util.Scanner;

```
class ThreeDObject {
public double wholeSurfaceArea()
{ return 0.0;
}
public double volume()
```

```
{ return 0.0;
}
}
class Box extends ThreeDObject
{ private double length;
private double width;
private double height;
public Box(double length, double width, double height)
{ this.length = length;
this.width = width;
this.height = height;
}
public double wholeSurfaceArea() {
return 2 * (length * width + width * height + height * length);
}
public double volume() {
return length * width * height;
}
}
class Cube extends ThreeDObject
{ private double side;
public Cube(double side)
{ this.side = side;
}
public double wholeSurfaceArea()
{ return 6 * side * side;
}
public double volume()
{ return side * side * side;
}
```

```
}
class Cylinder extends ThreeDObject
{ private double radius;
private double height;
public Cylinder(double radius, double height)
{ this.radius = radius;
this.height = height;
}
public double wholeSurfaceArea() {
return 2 * Math.PI * radius * (radius + height);
}
public double volume() {
return Math.PI * radius * radius * height;
}
}
class Cone extends ThreeDObject
{ private double radius;
private double height;
public Cone(double radius, double height)
{ this.radius = radius;
this.height = height;
}
public double wholeSurfaceArea() {
return Math.PI * radius * (radius + Math.sqrt(radius * radius + height * height));
}
public double volume() {
return (Math.PI * radius * radius * height) / 3.0;
}
}
public class TestThreeDObjects
```

```
{ public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.println("Enter dimensions for Box:");
System.out.print("Length: ");
double boxLength = scanner.nextDouble();
System.out.print("Width: ");
double boxWidth = scanner.nextDouble();
System.out.print("Height: ");
double boxHeight = scanner.nextDouble();
Box box = new Box(boxLength, boxWidth, boxHeight);
System.out.println("\nEnter dimensions for Cube:");
System.out.print("Side: ");
double cubeSide = scanner.nextDouble();
Cube cube = new Cube(cubeSide);
System.out.println("\nEnter dimensions for Cylinder:");
System.out.print("Radius: ");
double cylinderRadius = scanner.nextDouble();
System.out.print("Height: ");
double cylinderHeight = scanner.nextDouble();
Cylinder cylinder = new Cylinder(cylinderRadius, cylinderHeight);
System.out.println("\nEnter dimensions for Cone:");
System.out.print("Radius: ");
double coneRadius = scanner.nextDouble();
System.out.print("Height: ");
double coneHeight = scanner.nextDouble();
Cone cone = new Cone(coneRadius, coneHeight); System.out.println("\nResults:");
System.out.println("Box Surface Area: " + box.wholeSurfaceArea());
System.out.println("Box Volume: " + box.volume());
System.out.println("\nCube Surface Area: " + cube.wholeSurfaceArea());
System.out.println("Cube Volume: " + cube.volume());
System.out.println("\nCylinder Surface Area: " + cylinder.wholeSurfaceArea());
System.out.println("Cylinder Volume: " + cylinder.volume());
System.out.println("\nCone Surface Area: " + cone.wholeSurfaceArea());
System.out.println("Cone Volume: " + cone.volume());
scanner.close();
}
}
```



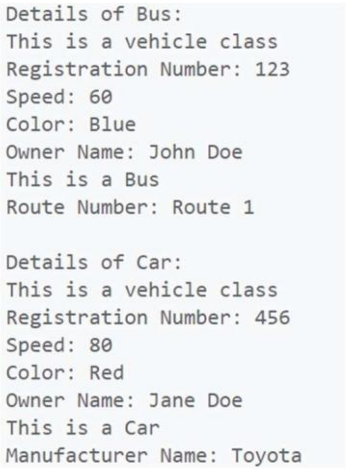
```
Length: 10
Width: 20
Height: 30

Enter dimensions for Cube:
Side: 40

Enter dimensions for Cylinder:
Radius: 50
Height: 60

Enter dimensions for Cone:
Radius: 70
Height: 20

Results:
Box Surface Area: 2200.0
Box Volume: 6000.0

Cube Surface Area: 9600.0
Cube Volume: 64000.0
```

6. 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.

Code:

```
public class TestVehicle {
}
class Vehicle {
protected String regnNumber;
protected int speed;
protected String color;
protected String ownerName;
public Vehicle(String regnNumber, int speed, String color, String ownerName)
{ this.regnNumber = regnNumber;
this.speed = speed;
this.color = color;
this.ownerName = ownerName;
}
public void showData()
{ System.out.println("This is a vehicle class");
System.out.println("Registration Number: " + regnNumber);
System.out.println("Speed: " + speed); System.out.println("Color: " + color);
System.out.println("Owner Name: " + ownerName);
}
}
class Bus extends Vehicle
{ private String routeNumber;
public Bus(String regnNumber, int speed, String color, String ownerName, String routeNumber)
{ super(regnNumber, speed, color, ownerName);
this.routeNumber = routeNumber;
}
public void showData()
{ super.showData();
System.out.println("This is a Bus");
System.out.println("Route Number: " + routeNumber);
}
}
class Car extends Vehicle {
private String manufacturerName;
 public Car(String regnNumber, int speed, String color, String ownerName, String manufacturerName) {
super(regnNumber, speed, color, ownerName);
this.manufacturerName = manufacturerName;
}
```

```
public void showData()
{ super.showData();
System.out.println("This is a Car");
System.out.println("Manufacturer Name: " + manufacturerName);
}
}
public class Test {
public static void main(String[] args) {
Bus myBus = new Bus("123", 60, "Blue", "John Doe", "Route 1");
Car myCar = new Car("456", 80, "Red", "Jane Doe", "Toyota");
System.out.println("Details of Bus:");
myBus.showData();
System.out.println("\nDetails of Car:");
myCar.showData();
}
}
```



```
Details of Bus:
This is a vehicle class
Registration Number: 123
Speed: 60
Color: Blue
Owner Name: John Doe
This is a Bus
Route Number: Route 1

Details of Car:
This is a vehicle class
Registration Number: 456
Speed: 80
Color: Red
Owner Name: Jane Doe
This is a Car
Manufacturer Name: Toyota
```

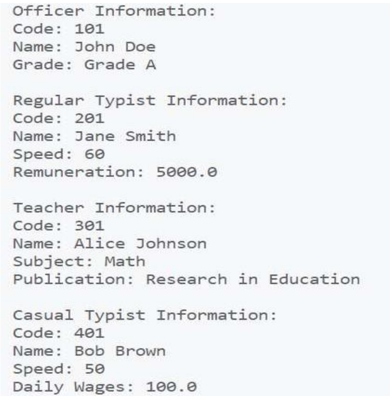
7. An educational institution maintains a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown below. Write all the classes and define the methods to create the database and retrieve individual information as and when needed. Write a driver program to test the classes. Staff (code, name) Officer (grade) is a Staff RegularTypist (remuneration) is a Typist Teacher (subject, publication) is a Staff Typist (speed) is a Staff CasualTypist (daily wages) is a Typist.

Code: class Staff

```
{ private int code;
private String name;
public Staff(int code, String name)
{ this.code = code;
this.name = name;
}
public int getCode()
{ return code;
}
public String getName()
{ return name;
}
}
class Typist extends Staff
{ private int speed;
public Typist(int code, String name, int speed)
{ super(code, name);
this.speed = speed;
}
public int getSpeed()
{ return speed;
}
}
class Officer extends Staff
{ private String grade;
public Officer(int code, String name, String grade)
{ super(code, name);
this.grade = grade;
}
public String getGrade()
```

```
{ return grade;
}
}
class RegularTypist extends Typist
{ private double remuneration;
public RegularTypist(int code, String name, int speed, double remuneration)
{ super(code, name, speed);
this.remuneration = remuneration;
}
public double getRemuneration()
{ return remuneration;
}
}
class Teacher extends Staff
{ private String subject;
private String publication;
public Teacher(int code, String name, String subject, String publication)
{ super(code, name);
this.subject = subject;
this.publication = publication;
}
public String getSubject()
{ return subject;
}
public String getPublication()
{ return publication;
}
}
class CasualTypist extends Typist
{ private double dailyWages;
```

```
public CasualTypist(int code, String name, int speed, double dailyWages)
{ super(code, name, speed);
this.dailyWages = dailyWages;
}
public double getDailyWages()
{ return dailyWages;
}
}
public class EducationalInstitution { public static void main(String[] args) {
Officer officer = new Officer(101, "John Doe", "Grade A");
RegularTypist regularTypist = new RegularTypist(201, "Jane Smith", 60, 5000.0);
Teacher teacher = new Teacher(301, "Alice Johnson", "Math", "Research in Education");
CasualTypist casualTypist = new CasualTypist(401, "Bob Brown", 50, 100.0);
System.out.println("Officer Information:");
System.out.println("Code: " + officer.getCode());
System.out.println("Name: " + officer.getName());
System.out.println("Grade: " + officer.getGrade());
System.out.println("\nRegular Typist Information:");
System.out.println("Code: " + regularTypist.getCode());
System.out.println("Name: " + regularTypist.getName());
System.out.println("Speed: " + regularTypist.getSpeed());
System.out.println("Remuneration: " + regularTypist.getRemuneration());
System.out.println("\nTeacher Information:");
System.out.println("Code: " + teacher.getCode());
System.out.println("Name: " + teacher.getName());
System.out.println("Subject: " + teacher.getSubject());
System.out.println("Publication: " + teacher.getPublication());
System.out.println("\nCasual Typist Information:");
System.out.println("Code: " + casualTypist.getCode());
System.out.println("Name: " + casualTypist.getName());
System.out.println("Speed: " + casualTypist.getSpeed());
System.out.println("Daily Wages: " + casualTypist.getDailyWages());
}
}
```



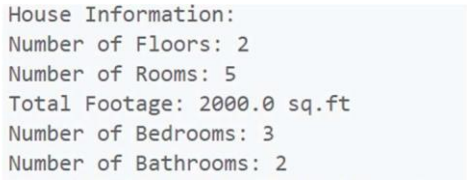
```
Officer Information:
Code: 101
Name: John Doe
Grade: Grade A

Regular Typist Information:
Code: 201
Name: Jane Smith
Speed: 60
Remuneration: 5000.0

Teacher Information:
Code: 301
Name: Alice Johnson
Subject: Math
Publication: Research in Education

Casual Typist Information:
Code: 401
Name: Bob Brown
Speed: 50
Daily Wages: 100.0
```

8. 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.

Code:

```
class Building {
protected int numberOfFloors;
protected int numberOfRooms;
protected double totalFootage;
public Building(int numberOfFloors, int numberOfRooms, double totalFootage)
{ this.numberOfFloors = numberOfFloors;
this.numberOfRooms = numberOfRooms;
this.totalFootage = totalFootage;
}
public void displayInfo() {
System.out.println("Number of Floors: " + numberOfFloors);
System.out.println("Number of Rooms: " + numberOfRooms);
System.out.println("Total Footage: " + totalFootage + " sq.ft");
}
}
class House extends Building
```

```
{ private int numberOfBedrooms;
private int numberOfBathrooms;
public House(int numberOfFloors, int numberOfRooms, double totalFootage, int
numberOfBedrooms, int numberOfBathrooms) {
super(numberOfFloors, numberOfRooms, totalFootage);
this.numberOfBedrooms = numberOfBedrooms; this.numberOfBathrooms =
numberOfBathrooms;
}
public void displayInfo()
{ super.displayInfo();
System.out.println("Number of Bedrooms: " + numberOfBedrooms);
System.out.println("Number of Bathrooms: " + numberOfBathrooms);
}
}
public class Main {
public static void main(String[] args) {
House myHouse = new House(2, 5, 2000.0, 3, 2);
System.out.println("House Information:");
myHouse.displayInfo();
}
}
```



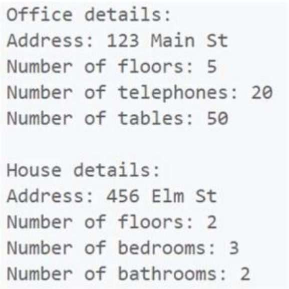
```
House Information:
Number of Floors: 2
Number of Rooms: 5
Total Footage: 2000.0 sq.ft
Number of Bedrooms: 3
Number of Bathrooms: 2
```

9. 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.

Code: class Building

```
{ protected String address;
protected int floors;
public Building(String address, int floors)
```

```
{ this.address = address;
this.floors = floors;
}
public void display()
{ System.out.println("Address: " + address);
System.out.println("Number of floors: " + floors);
}
}
class Office extends Building
{ private int telephones;
private int tables;
public Office(String address, int floors, int telephones, int tables)
{ super(address, floors);
this.telephones = telephones;
this.tables = tables;
}
public void display()
{ super.display();
System.out.println("Number of telephones: " + telephones);
System.out.println("Number of tables: " + tables);
}
}
class House extends Building
{ private int bedrooms;
private int bathrooms;
public House(String address, int floors, int bedrooms, int bathrooms)
{ super(address, floors);
this.bedrooms = bedrooms; this.bathrooms = bathrooms;
}
public void display() { super.display();
System.out.println("Number of bedrooms: " + bedrooms);
System.out.println("Number of bathrooms: " + bathrooms);
}
}
public class BuildingDemo {
public static void main(String[] args) {
Office office = new Office("123 Main St", 5, 20, 50);
House house = new House("456 Elm St", 2, 3, 2);
System.out.println("Office details:");
office.display();
System.out.println("\nHouse details:");
house.display();
}
}
```

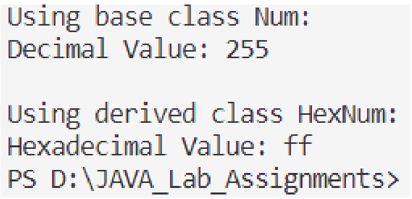


```
Office details:
Address: 123 Main St
Number of floors: 5
Number of telephones: 20
Number of tables: 50

House details:
Address: 456 Elm St
Number of floors: 2
Number of bedrooms: 3
Number of bathrooms: 2
```

10. 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.

Code:

```
class Num
{ protected int number;
public Num(int number)
{ this.number = number;
}
public void shownum()
{ System.out.println("Decimal Value: " + number);
}
}
class HexNum extends Num
{ public HexNum(int number)
{super(number);
}
public void shownum() {
System.out.println("Hexadecimal Value: " + Integer.toHexString(number));
}
}
public class NumDemo {
public static void main(String[] args)
{ Num numObj = new Num(255);
System.out.println("Using base class Num:");
numObj.shownum();
HexNum hexNumObj = new HexNum(255);
System.out.println("\nUsing derived class HexNum:");
hexNumObj.shownum();
}
}
```



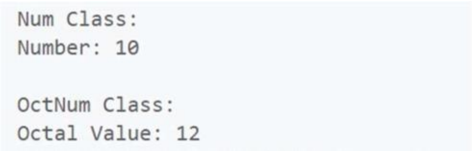
```
Using base class Num:
Decimal Value: 255

Using derived class HexNum:
Hexadecimal Value: ff
PS D:\JAVA_Lab_Assignments>
```

11. 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.

Code: class Num

```
{ protected int number; public Num(int number) {
```

```
this.number = number;
}
public void showNum()
{ System.out.println("Number: " + number);
}
}
class OctNum extends Num
{ public OctNum(int number)
{super(number);
}
public void showNum()
{System.out.println("Octal Value: " + Integer.toOctalString(number));
}
}
public class Main {
public static void main(String[] args)
{ Num num = new Num(10);
OctNum octNum = new OctNum(10);
System.out.println("Num Class:");
num.showNum();
System.out.println("\nOctNum Class:");
octNum.showNum();
}
}
```



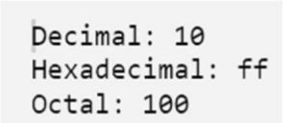
```
Num Class:
Number: 10

OctNum Class:
Octal Value: 12
```

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

Code: class Num

```
{ int number;
```

```
public Num(int number)
{ this.number = number;
}
public void shownum()
{ System.out.println("Decimal: " + number);
}
}
class HexNum extends Num
{ public HexNum(int number)
{super(number);
}
public void shownum()
{System.out.println("Hexadecimal: " + Integer.toHexString(number));
}
}
class OctNum extends Num
{ public OctNum(int number)
{super(number);
}
public void shownum()
{System.out.println("Octal: " + Integer.toOctalString(number));
}
}
public class Main {
public static void main(String[] args)
{ Num num1 = new Num(10);
num1.shownum();
HexNum hex_num = new HexNum(255);
hex_num.shownum();
OctNum oct_num = new OctNum(64);
oct_num.shownum();
}
}
```

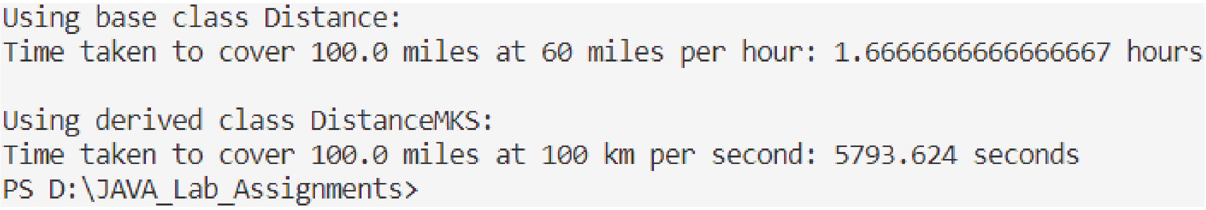


```
Decimal: 10
Hexadecimal: ff
Octal: 100
```

13. 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.

Code:

```
class Distance {
protected double distanceMiles;
public Distance(double distanceMiles)
{ this.distanceMiles = distanceMiles;
}
public void travelTime() {
double speedMilesPerHour = 60.0;
double timeHours = distanceMiles / speedMilesPerHour;
System.out.println("Time taken to cover " + distanceMiles + " miles at 60 miles per hour: " + timeHours + " hours");
}
}
class DistanceMKS extends Distance {
public DistanceMKS(double distanceMiles)
{ super(distanceMiles);
}
public void travelTime() {
double speedKilometersPerSecond = 100.0 / 3600.0;
double distanceKilometers = distanceMiles * 1.60934;
double timeSeconds = distanceKilometers / speedKilometersPerSecond;
System.out.println("Time taken to cover " + distanceMiles + " miles at 100 km per second: " + timeSeconds + " seconds");
}
}
public class DistanceDemo {
public static void main(String[] args)
{ Distance distanceObj = new Distance(100);
System.out.println("Using base class Distance:");
distanceObj.travelTime();
DistanceMKS distanceMKSObj = new DistanceMKS(100);
System.out.println("\nUsing derived class DistanceMKS:");
distanceMKSObj.travelTime();
}
}
```



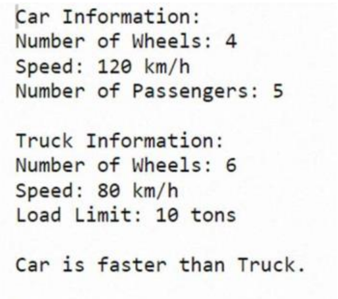
```
Using base class Distance:
Time taken to cover 100.0 miles at 60 miles per hour: 1.6666666666666667 hours

Using derived class DistanceMKS:
Time taken to cover 100.0 miles at 100 km per second: 5793.624 seconds
PS D:\JAVA_Lab_Assignments>
```

14. 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.

Code:

```
class Vehicle {
protected int numberOfWheels;
protected int speed;
public Vehicle(int numberOfWheels, int speed)
{ this.numberOfWheels = numberOfWheels;
this.speed = speed;
public int getSpeed()
{ return speed;
public void displayInfo() {
System.out.println("Number of Wheels: " + numberOfWheels);
System.out.println("Speed: " + speed + " km/h");
}
}
class Car extends Vehicle {
private int numberOfPassengers;
public Car(int numberOfWheels, int speed, int numberOfPassengers)
{ super(numberOfWheels, speed);
this.numberOfPassengers = numberOfPassengers;
}
public void displayInfo()
{ super.displayInfo();
System.out.println("Number of Passengers: " + numberOfPassengers);
}
}
class Truck extends Vehicle
{ private int loadLimit;
public Truck(int numberOfWheels, int speed, int loadLimit)
{ super(numberOfWheels, speed);
this.loadLimit = loadLimit;
}
public void displayInfo()
{ super.displayInfo();
System.out.println("Load Limit: " + loadLimit + " tons");
}
}
public class Main {
public static void main(String[] args)
{ Car myCar = new Car(4, 120, 5);
Truck myTruck = new Truck(6, 80, 10);
System.out.println("Car Information:");
```

```
myCar.displayInfo();
System.out.println("\nTruck Information:");
myTruck.displayInfo();
if (myCar.getSpeed() > myTruck.getSpeed())
{ System.out.println("\nCar is faster than Truck.");
}
else if (myCar.getSpeed() < myTruck.getSpeed())
{ System.out.println("\nTruck is faster than Car.");
} else {
System.out.println("\nCar and Truck have the same speed.");
}
}
```

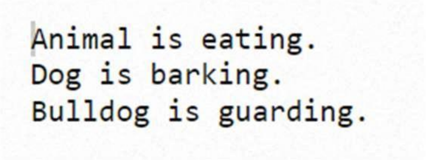


```
Car Information:
Number of Wheels: 4
Speed: 120 km/h
Number of Passengers: 5

Truck Information:
Number of Wheels: 6
Speed: 80 km/h
Load Limit: 10 tons

Car is faster than Truck.
```

15. Write a Java program to explain "multilevel inheritance."

Code: class Animal

```
{ void eat() {
System.out.println("Animal is eating.");
}
}
class Dog extends Animal
{ void bark()
{System.out.println("Dog is barking.");
}
Dog()
{super();
}
}
class Bulldog extends Dog
{ void guard()
{System.out.println("Bulldog is guarding.");
}
Bulldog()
{ super();
}
}
public class Main {
public static void main(String[] args)
{ Bulldog myDog = new Bulldog();
myDog.eat();
myDog.bark();
myDog.guard();
}
}
```



```
Animal is eating.
Dog is barking.
Bulldog is guarding.
```