

INDEX

S.no	Topics	Page No
1.	Decision Making Statement	1
2.	While and do while basic program	24
3.	For loop basic program	38
4.	Array basic program	69
5.	Function basic program	86
6.	Class and Object	95
7.	Constructor	98
8.	Inheritance and types	102
9.	Polymorphism	108
10.	Exception Handling	117
11.	String	120
12.	Array List	126
13.	Linked List	133
14.	Hash Map	137
15.	Hash Set	142
16.	Linked Hash Set	143
17.	IO Package	144
18.	JDBC connectivity	158

Example for types of variable:

```
public class A {  
    int data=50;//instance variable  
    static int m=100;//static variable  
    void method()  
    {  
        int n=90;//local variable  
    }  
    public static void main(String args[])  
    {  
        }  
} //end of class
```

Control Statements:

1. Decision-making statement:

- if statement
 - simple if statement
 - if-else statement
 - if-else-if ladder
 - nested if-statement
- switch statement

Simple if statement:

```
if(condition)  
{  
    statement ;  
}
```

If-else statement:

Syntax:

```
if(condition)
{
    statement1;
}
else
{
    statement2;
}
```

If-else-if ladder

Syntax:

```
if(condition 1) {
    statement;
}
else if(condition 2) {
    statement;
}
else {
    statement;
}
```

Nested-if-else:

Syntax:

```
if(condition 1) {
    statement;
if(condition 2) {
    statement;
}
else{
    statement;
```

```
}  
}
```

Switch statement:

Syntax:

```
switch (expression){  
    case value1:  
        statement1;  
        break;  
    .  
    .  
    .  
    case valueN:  
        statementN;  
        break;  
    default:  
        default statement;  
}
```

Example Programs

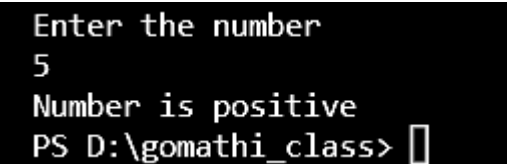
1. Write a Java program to get a number from the user and print whether it is positive or negative.

Program:

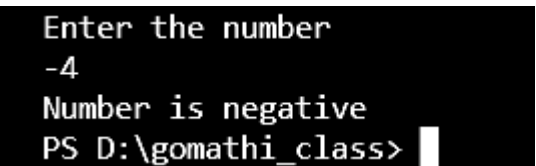
```
import java.util.Scanner;

public class PositiveOrNegative {
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the number");
        int number = obj.nextInt();
        if (number>0){
            System.out.println("Number is positive");
        }
        else{
            System.out.println("Number is negative");
        }
    }
}
```

Output:

A screenshot of a terminal window showing the execution of the Java program. The prompt 'Enter the number' is followed by the user input '5'. The program then outputs 'Number is positive'. The command prompt 'PS D:\gomathi_class>' is visible at the bottom.

```
Enter the number
5
Number is positive
PS D:\gomathi_class>
```

A screenshot of a terminal window showing the execution of the Java program. The prompt 'Enter the number' is followed by the user input '-4'. The program then outputs 'Number is negative'. The command prompt 'PS D:\gomathi_class>' is visible at the bottom.

```
Enter the number
-4
Number is negative
PS D:\gomathi_class>
```

2. Take three numbers from the user and print the greatest number.

Program:

```
import java.util.Scanner;
public class GreatestNumber{
    public static void main(String args[])
    {
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the value of a");
        int a=obj.nextInt();
        System.out.println("Enter the value of b");
        int b=obj.nextInt();
        System.out.println("Enter the value of c");
        int c=obj.nextInt();
        if (a>b && a>c)
        {
            System.out.println("A is greater");
        }
        else if(b>c && b>a)
        {
            System.out.println("B is greater");
        }
        else
        {
            System.out.println("C is greater");
        }
    }
}
```

Output:

```
Enter the value of a
30
Enter the value of b
20
Enter the value of c
10
A is greater
```

3. Take values of the length and breadth of a rectangle from the user and check if it is square or not.

Program:

```
import java.util.Scanner;
public class CheckRectangleorSquare {
    public static void main(String args[]){
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the length");
        int length=obj.nextInt();
        System.out.println("Enter the breadth");
        int breadth=obj.nextInt();
        if (length==breadth){
            System.out.println("square");
        }
        else{
            System.out.println("Rectangle");
        }
    }
}
```

Output:

```
Enter the length
10
Enter the breadth
20
Rectangle
```

```
Enter the length
20
Enter the breadth
20
square
```


4. Write a Java program that keeps a number from the user and generates an integer between 1 and 7 and displays the name of the weekday.

Program:

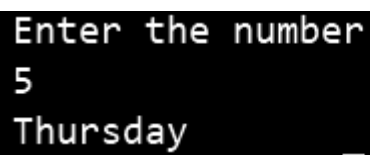
```
import java.util.Scanner;

public class Weekdays{
    public static void main(String args[]){
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the number");
        int input= obj.nextInt();
        switch(input){
            case 1:
                System.out.println("Sunday");
                break;
            case 2:
                System.out.println("Monday");
                break;
            case 3:
                System.out.println("Tuesday");
                break;
            case 4:
                System.out.println("Wednesday");
                break;
            case 5:
                System.out.println("Thursday");
                break;
            case 6:
                System.out.println("Friday");
                break;
```

```
        case 7:
            System.out.println("Saturday");
            break;
        default:
            System.out.println("enter the number
between 1 to 7");
    }

}
}
```

Output:

A screenshot of a terminal window with a black background and white text. It shows the output of a Java program: "Enter the number", followed by the user input "5", and the resulting output "Thursday".

```
Enter the number
5
Thursday
```

5. A company decided to give a bonus of 5% to an employee if his/her year of service is more than 5 years. Ask users for their salary and year of service and print the net bonus amount.

Program:

```
import java.util.Scanner;

public class Salary {
    public static void main(String args[]){
        float bonus;
        float salary;
        int year;
        Scanner obj=new Scanner(System.in);
        System.out.println("enter the number of
years");
        year=obj.nextInt();
        System.out.println("enter the salary");
        salary=obj.nextInt();
        if (year>5){
            bonus= 0.05f*salary;
            System.out.println("Your Salary is:"+salary);

            System.out.println("Your Bonus is:"+bonus);
            System.out.println(" Your Net Bonus with Salary
is: "+(salary+bonus));
        }
        else{
            System.out.println("No Bonus");
        }
    }
}
```

}

Output:

```
C:\java\jre_1.7\bin>java -cp .\gomathi_class\classes\bin\gomathi.class  
enter the number of years  
6  
enter the salary  
50000  
Your Salary is:50000.0  
Your Bonus is:2500.0  
Your Net Bonus with Salary is:52500.0
```

```
enter the number of years  
4  
enter the salary  
30000  
No Bonus  
PS D:\gomathi_class>
```

6. A school has the following rules for the grading system:

- a. Below 25 - F
- b. 25 to 45 - E
- c. 45 to 50 - D
- d. 50 to 60 - C
- e. 60 to 80 - B
- f. Above 80 - A

Ask users to enter marks and print the corresponding grade.

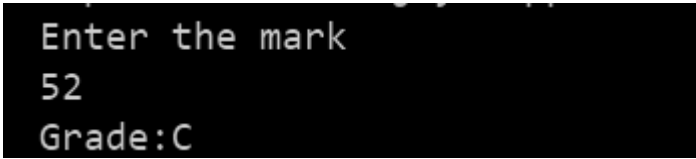
Program:

```
import java.util.Scanner;

public class Grading {
    public static void main(String args[]){
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the mark");
        int mark = obj.nextInt();
        if(mark>80){
            System.out.println("Grade:A");
        }
        else if (mark>=60 && mark<=80){
            System.out.println("Grade:B");
        }
        else if (mark>=50 && mark<60){
            System.out.println("Grade:C");
        }
        else if (mark >=45 && mark <50){
            System.out.println("Grade:D");
        }
        else if (mark>=25 && mark <45){
            System.out.println("Grade :E");
        }
    }
}
```

```
    }  
    else{  
        System.out.println("Grade:F");  
    }  
}  
}
```

Output:



```
Enter the mark  
52  
Grade:C
```

7. Write a program to find the number of days in a month?

Program:

```
import java.util.Scanner;

public class NumberofDays {
    private static String monthName;

    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        monthName="";
        int number_Of_DaysInMonth = 0;
        System.out.println("enter a month in
number:");
        int month = obj.nextInt();
        System.out.println("enter a year");
        int year = obj.nextInt();
        switch (month) {
            case 1:
                monthName = "January";
                number_Of_DaysInMonth = 31;
                break;
            case 2:
                monthName = "February";
                if ((year % 400 == 0) || ((year % 4
== 0) && (year % 100 != 0))) {
                    number_Of_DaysInMonth = 29;
                } else {
                    number_Of_DaysInMonth = 28;
                }
            }
        }
    }
}
```

```
        break;
case 3:
    monthName = "March";
    number_Of_DaysInMonth = 31;
    break;
case 4:
    monthName = "April";
    number_Of_DaysInMonth = 30;
    break;
case 5:
    monthName = "May";
    number_Of_DaysInMonth = 31;
    break;
case 6:
    monthName = "June";
    number_Of_DaysInMonth = 30;
    break;
case 7:
    monthName = "July";
    number_Of_DaysInMonth = 31;
    break;
case 8:
    monthName = "August";
    number_Of_DaysInMonth = 31;
    break;
case 9:
    monthName = "September";
    number_Of_DaysInMonth = 30;
    break;
case 10:
    monthName = "October";
    number_Of_DaysInMonth = 31;
```



```

        break;
    case 11:
        monthName = "November";
        number_Of_DaysInMonth = 30;
        break;
    case 12:
        monthName = "December";
        number_Of_DaysInMonth = 31;
    }
    System.out.print("In " + monthName + " " +
year + " has " + number_Of_DaysInMonth + " days\n");
    }
}

```

Output:

```

enter a month
2
enter a year
2008
February 2008 has 29 days

```

8. Write a java program that takes a year from the user and print whether that year is a leap year or not?

Program:

```
import java.util.Scanner;

public class LeapYearOrNot {
    public static void main(String args[]){
        Scanner obj= new Scanner(System.in);
        System.out.println("Enter the year");
        int year = obj.nextInt();
        if(year%100==0){

            if(year%400==0){
                System.out.println("Entered year is
leap year");
            }
            else{
                System.out.println("Enter year is not
a leap year");
            }
        }
        else{
            if(year%4==0){
                System.out.println("Entered year is
leap year ");
            }
            else{
```

```
                System.out.println("Entered year is  
not a leap year");  
            }  
        }  
    }  
}
```

Output:

```
Enter the year  
2006  
Entered year is not a leap year
```

```
Enter the year  
2016  
Entered year is leap year
```

9. Write a program to check whether a number is divisible by 7 or not.

Program:

```
import java.util.Scanner;

public class Divisible7 {
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the number");
        int num=obj.nextInt();
        if (num%7==0){
            System.out.println("Number is divisible
by 7");
        }
        else{
            System.out.println("Number is not
divisible by 7");
        }
    }
}
```

Output:

```
Enter the number
50
Number is not divisible by 7
```

```
Enter the number
49
Number is divisible by 7
```

10. Accept three sides of a triangle and check whether it is an equilateral, isosceles, or scalene triangle.

Note:

An equilateral triangle is a triangle with all three sides equal.

A scalene triangle is a triangle that has three unequal sides.

An isosceles triangle is a triangle with (at least) two equal sides.

Program:

```
import java.util.Scanner;

public class TriangleCheck {
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter first side of
triangle");
        int s1=obj.nextInt();
        System.out.println("Enter second side of
triangle");
        int s2=obj.nextInt();
        System.out.println("Enter third side of
triangle");
        int s3=obj.nextInt();
        if(s1==s2 && s2==s3){
            System.out.println("Equilateral
triangle");
        }
        if ((s1==s2 && s2!=s3) || (s2==s3 && s2!=s1)
|| (s1==s3 && s1!=s2)){
            System.out.println("Isosceles triangle");
        }
    }
}
```

```
    }  
    if(s1!=s2 && s1!=s3 && s2!=s3)  
    {  
        System.out.println("Scalene triangle");  
    }  
}  
}
```

Output:

```
Enter first side of triangle  
20  
Enter second side of triangle  
30  
Enter third side of triangle  
60  
Scalene triangle
```

11. Write a program to check whether a number entered is a three-digit number or not.

Program:

```
import java.util.Scanner;

public class CheckThreeDigitNumber {
    public static void main(String args[]){
        Scanner obj= new Scanner(System.in);
        System.out.println("Enter any number");
        int number=obj.nextInt();
        if(number>=100 && number<=999){
            System.out.println("it is a three digit
number");
        }
        else{
            System.out.println("It is not a three
digit number");
        }
    }
}
```

Output:

```
Enter any number
45
It is not a three digit number
```

```
Enter any number
456
it is a three digit number
```

12. Write a program to check the user entered age is eligible to vote.

Program:

```
import java.util.Scanner;

public class Voting {
    public static void main(String args[]){
        Scanner obj= new Scanner(System.in);
        System.out.println("Enter the age");
        int age = obj.nextInt();
        if(age>=18){
            System.out.println("eligible to vote");
        }
        else{
            System.out.println("not eligible to
vote");
        }
    }
}
```

Output:

```
Enter the age
18
eligible to vote
```

```
Enter the age
12
not eligible to vote
```


2.Looping statement:

- while loop
- do-while loop
- for loop
- for each loop

while loop:

syntax:

```
while(condition){  
    looping statement;  
}
```

do-while loop:

syntax:

```
do{  
    looping statement;  
}while(condition);
```

for loop:

syntax:

```
for(initialization;condition;increment/decrement){  
    looping statement;  
}
```

for each loop:

syntax:

```
for(data_type var : array_name/collection_name){  
    looping statement;
```

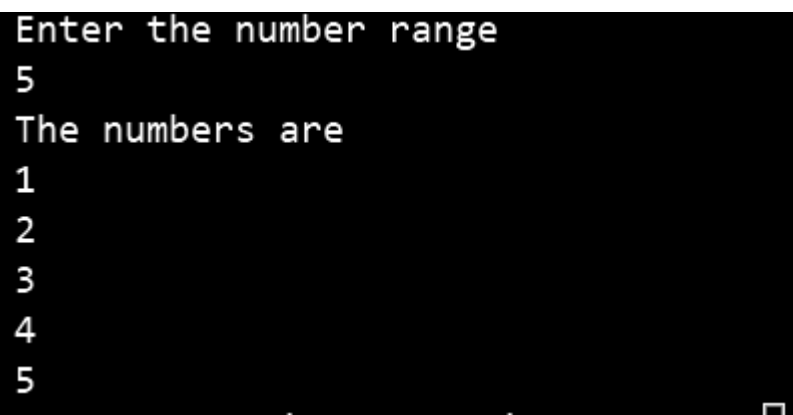
13. Write a program to print the following using a while loop for the user-entered range of natural numbers.

Program:

```
import java.util.Scanner;

public class Natural{
    public static void main(String args[]){
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the number range");
        int range=obj.nextInt();
        System.out.println("The numbers are");
        int i=1;
        while(i<=range){
            System.out.println(i);
            i+=1;
        }
    }
}
```

Output:



```
Enter the number range
5
The numbers are
1
2
3
4
5
```

14. Write a program to print the following using a while loop for the user entered the range of even numbers.

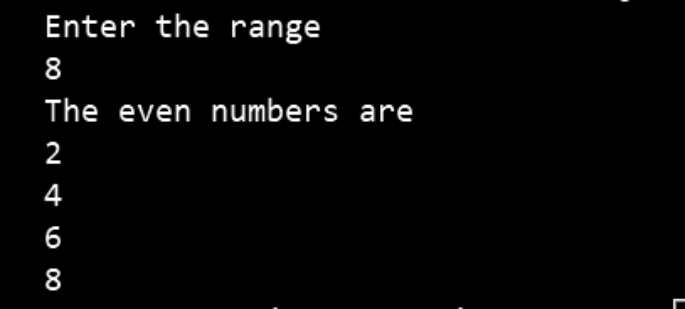
Program:

```
import java.util.Scanner;

public class EvenNumber {
    public static void main(String args[]){
        Scanner obj= new Scanner(System.in);{
            System.out.println("Enter the range");
            int range=obj.nextInt();
            System.out.println("The even numbers
are");

            int i=1;
            while(i<=range){
                if(i%2==0){
                    System.out.println(i);
                }
                i++;
            }
        }
    }
}
```

Output:

A screenshot of a terminal window with a black background and white text. The text shows the program's execution: it prompts 'Enter the range', the user enters '8', it then prints 'The even numbers are', followed by a list of even numbers from 2 to 8 on separate lines.

```
Enter the range
8
The even numbers are
2
4
6
8
```

15. Write a program to print for the user entered range integer of their squares using a while loop.

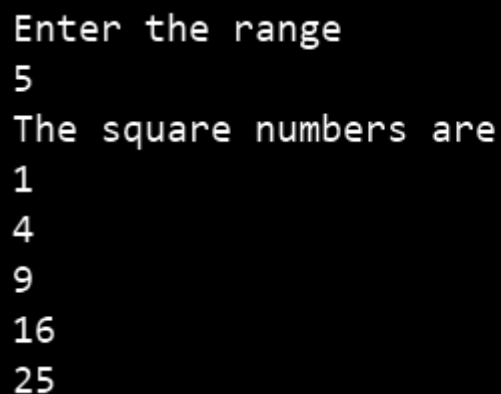
Program:

```
import java.util.Scanner;

public class Square {
    public static void main(String args[]){
        Scanner obj= new Scanner(System.in);{
            System.out.println("Enter the range");
            int range=obj.nextInt();
            System.out.println("The square numbers
are");

            int i=1;
            while(i<=range){
                System.out.println(i*i);
                i+=1;
            }
        }
    }
}
```

Output:

A screenshot of a terminal window with a black background and white text. The output shows the program's execution: it prompts 'Enter the range', the user enters '5', it then prints 'The square numbers are', and finally lists the squares of integers from 1 to 5: '1', '4', '9', '16', and '25' on separate lines.

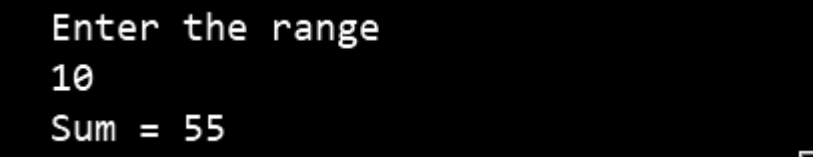
```
Enter the range
5
The square numbers are
1
4
9
16
25
```

16. Write a program to print the sum of the first n user entered range Natural numbers.

Program:

```
import java.util.Scanner;
public class SumOfNatural {
    public static void main(String args[]){
        Scanner obj= new Scanner(System.in);{
            System.out.println("Enter the range");
            int range=obj.nextInt();
            int i=1,sum=0;
            do{
                sum+=i;
                i+=1;
            }while(i<=range);
            System.out.println("Sum = "+sum);
        }
    }
}
```

Output:

A screenshot of a terminal window with a black background and white text. It shows the output of the program: 'Enter the range', followed by the user input '10', and the final result 'Sum = 55'.

```
Enter the range
10
Sum = 55
```

17. Write a program to print a table of a number entered from the user.

Program:

```
import java.util.Scanner;
public class Table{
    public static void main(String args[]){
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the number range");
        int range = obj.nextInt();
        System.out.println("Enter the number");
        int number=obj.nextInt();
        int i = 1;
        do{
            System.out.println(i+"x"+number+"="+i*number);
            i+=1;
        }while(i<=range);
    }
}
```

Output:

```
Enter the number range
10
Enter the number
3
1x3=3
2x3=6
3x3=9
4x3=12
5x3=15
6x3=18
7x3=21
8x3=24
9x3=27
10x3=30
```

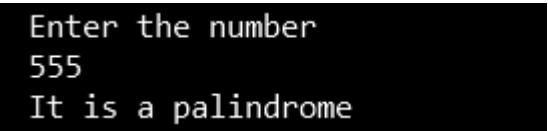
18. Write a program to check whether a user entered number is a palindrome or not.

Program:

```
import java.util.Scanner;

public class Palindrome {
    public static void main(String args[]) {
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the number");
        int number = obj.nextInt();
        int result = 0;
        int originalNumber = number;
        while (number > 0) {
            int rem = number % 10;
            result = (result * 10) + rem;
            number /= 10;
        }
        if (originalNumber == result) {
            System.out.println("It is a palindrome");
        } else {
            System.out.println("It is not a
palindrome");
        }
    }
}
```

Output:

A screenshot of a terminal window with a black background and white text. It shows the output of the Java program: "Enter the number", followed by the user input "555", and the program's response "It is a palindrome".

```
Enter the number
555
It is a palindrome
```

```
Enter the number  
453  
It is not a palindrome
```


19. Write a program to check whether the user entered a three-digit number is an Armstrong number or not.

Program:

```
import java.util.Scanner;
public class Armstrong {
    public static void main(String args[]){
        Scanner obj= new Scanner(System.in);
        System.out.println("Enter the three digit
number");
        int number= obj.nextInt();
        int result=0;
        int originalNumber=number;
        while((number>99) && (number < 1000)){
            int rem=number%10;
            result+=rem*rem*rem;
            number/=10;
        }
        if(originalNumber==result){
            System.out.println("It is a Armstrong");
        }
        else{
            System.out.println("It is not a
Armstrong");
        }
    }
}
```

Output:

```
Enter the number  
534  
It is not a Armstrong
```

```
Enter the number  
407  
It is a Armstrong
```

20. Write a program to find the product of the digits of the number entered by the user.

Program:

```
import java.util.Scanner;

public class Product {
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the number");
        int number=obj.nextInt();
        int lastDigit=0;
        int product=1;
        while(number!=0){
            lastDigit=number%10;
            product=product*lastDigit;
            number=number/10;
        }
        System.out.println("Product of the
number:"+product);
    }
}
```

Output:

```
Enter the number
342
Product of the number:24
```

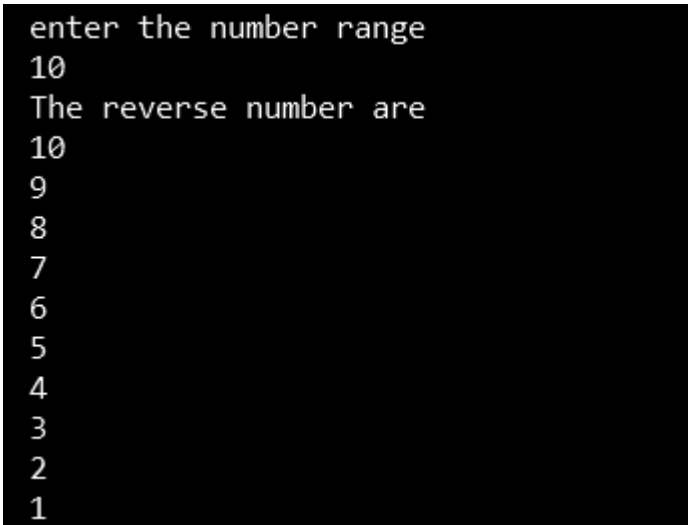
21. Write a program to print the first n user-entered natural numbers in reverse order using a while loop.

Program:

```
import java.util.Scanner;

public class Reverse {
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("enter the number range");
        int number=obj.nextInt();
        System.out.println("The reverse number are");
        while(number>=1){
            System.out.println(number);
            number-=1;
        }
    }
}
```

Output:

A screenshot of a terminal window with a black background and white text. The output shows the program's execution: it prompts for a number range, receives '10', and then prints the numbers from 10 down to 1 in reverse order.

```
enter the number range
10
The reverse number are
10
9
8
7
6
5
4
3
2
1
```

22. Write a Program to print all the characters in the string 'OCEAN' using a while loop.

Program:

```
public class Ocean {  
    public static void main(String args[]){  
        String str = "ocean";  
        int i=0;  
        while(i<str.length()){  
            System.out.println(str.charAt(i));  
            i+=1;  
        }  
    }  
}
```

Output:



```
o  
c  
e  
a  
n
```

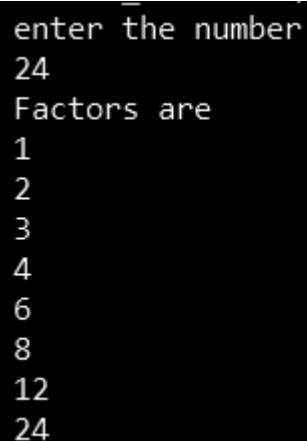
23. Write a program to print all the factors of a user-entered number using a while loop.

Program:

```
import java.util.Scanner;

public class FindFactors {
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("enter the number");
        int number=obj.nextInt();
        System.out.println("Factors are");
        int i=1;
        while(i<=number){
            if(number%i==0){
                System.out.println(i);
            }
            i+=1;
        }
    }
}
```

Output:

A screenshot of a terminal window showing the output of the Java program. The text is as follows:

```
enter the number
24
Factors are
1
2
3
4
6
8
12
24
```

3. For loop basic programs

24. Write a program to find the factorial for the user-entered number.

Program:

```
import java.util.Scanner;

public class Factorial {
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("enter the number");
        int number=obj.nextInt();
        int fact=1;
        for(int i=1;i<=number;i++){
            fact=fact*i;
        }
        System.out.println("Factorial number is:
"+fact);
    }
}
```

Output:

```
enter the number
5
Factorial number is: 120
```

25. Write a program to check whether the user entered number is a prime number or not.

Program:

```
public class PrimeOrNot{
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("enter the number");
        int number=obj.nextInt();
        int count=0;
        if (number==0 || number==1){
            System.out.println("it is not prime
number");
        }
        if(number >= 2){
            for(int i=2;i<=number;i++){
                if(number%i==0){
                    count++;
                }
            }
        }

        if(count==1){
            System.out.println("it is prime
number");
        }
        else{
            System.out.println("it is not prime
number");
        }
    }
}
```


Output:

```
enter the number  
5  
it is a prime number
```

```
enter the number  
6  
it is not a prime number
```

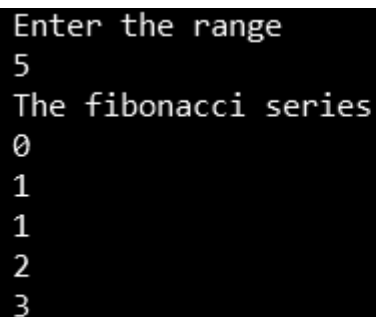
26. Write a program to find the Fibonacci series for the user-entered number.

Program:

```
import java.util.Scanner;

public class Fibonacci {
    public static void main(String args[]) {
        Scanner obj = new Scanner(System.in);
        int a = -1;
        int b = 1;
        System.out.println("Enter the range");
        int input = obj.nextInt();
        System.out.println("The fibonacci series");
        for (int i = 0; i < input; i++) {
            int c = a + b;
            a = b;
            b = c;
            System.out.println(c);
        }
    }
}
```

Output:

A screenshot of a terminal window with a black background and white text. The text shows the program's execution: it prompts 'Enter the range', the user enters '5', it prompts 'The fibonacci series', and then it prints the sequence '0', '1', '1', '2', '3' on separate lines.

```
Enter the range
5
The fibonacci series
0
1
1
2
3
```

27. Write a program to print the below pattern.

```
*
*  *
*  *  *
*  *  *  *
```

Program:

```
public class Pattern {
    public static void main(String args[]) {
        for (int i = 1; i < 5; i++) {
            for (int j = 0; j < i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

Output:



```
*
* *
* * *
* * * *
```

28. Write a program to print the below pattern based on the user entering the number of rows.

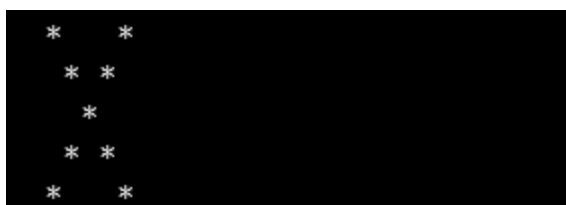
```
*   *
 *  *
  *
 *  *
*   *
```

Program:

```
public class Pattern{
    public static void main(String args[]) {
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 5; j++) {
                if (i == j || j == 5 - 1 - i) {
                    System.out.print("*");

                } else {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
    }
}
```

Output:



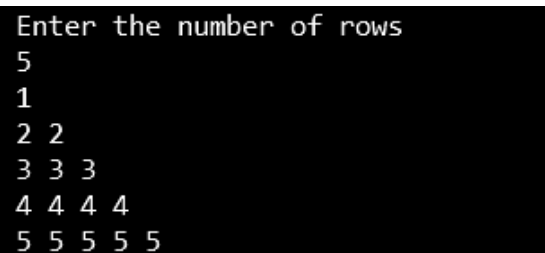
29. Write a program to print the below pattern based on the user entered a number of rows.

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

Program:

```
import java.util.Scanner;
public class Pattern {
    public static void main(String args[]) {
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows = obj.nextInt();
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(i + " ");
            }
            System.out.println();
        }
    }
}
```

Output:

A screenshot of a terminal window with a black background and white text. It shows the prompt 'Enter the number of rows' followed by the user input '5'. Below this, the program's output is displayed as a pattern of numbers: '1', '2 2', '3 3 3', '4 4 4 4', and '5 5 5 5 5' on separate lines.

```
Enter the number of rows
5
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

30. Write a program to print the below pattern based on the user entered a number of rows.

```
1 1 1 1 1 1
2 2 2 2 2 2
3 3 3 3 3 3
4 4 4 4 4 4
5 5 5 5 5 5
```

Program:

```
import java.util.Scanner;
public class Pattern{
    public static void main(String args[]) {
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows = obj.nextInt();
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows; j++) {
                System.out.print(i + " ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
Enter the number of rows
5
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5
```

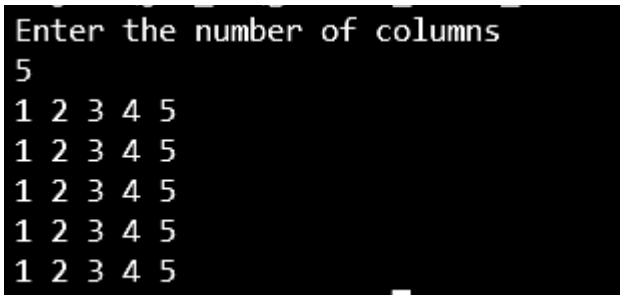
31. Write a program to print the below pattern based on the user entered a number of columns.

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

Program:

```
import java.util.Scanner;
public class Pattern{
    public static void main(String args[]) {
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the number of
columns");
        int rows = obj.nextInt();
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows; j++) {
                System.out.print(j + " ");
            }
            System.out.println();
        }
    }
}
```

Output:

A screenshot of a terminal window with a black background and white text. It shows the program's execution: a prompt 'Enter the number of columns' followed by the user input '5'. Below this, the program prints a 5x5 grid of numbers from 1 to 5, with spaces between the numbers in each row.

```
Enter the number of columns
5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

32. Write a program to print the below pattern based on the user entered a number of rows.

```
A A A A A
B B B B B
C C C C C
D D D D D
E E E E E
```

Program:

```
import java.util.Scanner;
public class Pattern {
    public static void main(String args[]) {
        int letter=64;
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows = obj.nextInt();
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows; j++) {
                System.out.print((char) (letter+i)+ "
");
            }
            System.out.println();
        }
    }
}
```


Output:

```
Enter the number of rows
5
A A A A A
B B B B B
C C C C C
D D D D D
E E E E E
```

33. Write a program to print the below pattern based on the user entering a number of columns.

```
A B C D E
A B C D E
A B C D E
A B C D E
A B C D E
```

Program:

```
import java.util.Scanner;
public class Pattern {
    public static void main(String args[]) {
        int letter=64;
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows = obj.nextInt();
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows; j++) {
                System.out.print((char) (letter+j)+ "
");
            }
            System.out.println();
        }
    }
}
```

Output:

```
Enter the number of rows
5
A B C D E
A B C D E
A B C D E
A B C D E
A B C D E
```

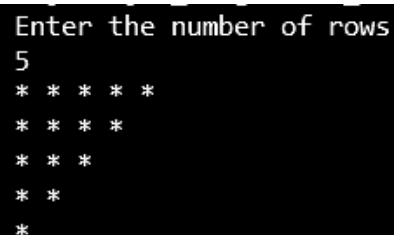
34. Write a program to print the below pattern based on the user entering a number of rows.

```
* * * *
* * *
* *
*
```

Program:

```
public class Pattern {
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows=obj.nextInt();
        for(int i=rows;i>0;i--){
            for(int j=0;j<i;j++){
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

Output:

A screenshot of a terminal window showing the program's output. The prompt 'Enter the number of rows' is followed by the user input '5'. Below this, the program prints a pattern of asterisks: the first row has 5 asterisks, the second has 4, the third has 3, the fourth has 2, and the fifth has 1. The asterisks are separated by a space.

```
Enter the number of rows
5
* * * * *
* * * *
* * *
* *
*
```

35. Write a program to print the below pattern based on the user entering a number of rows.

```
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

Program:

```
import java.util.Scanner;
public class Pattern {
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows=obj.nextInt();
        for(int i=rows;i>=1;i--){
            for(int j=1;j<=i;j++){
                System.out.print(i+" ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
Enter the number of rows
6
6 6 6 6 6 6
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

36. Write a program to print the below pattern based on the user entered a number of rows.

```
A
B C
D E F
G H I J
K L M N O
```

Program:

```
import java.util.Scanner;
public class Pattern{
    public static void main(String args[]){
        Scanner obj= new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows = obj.nextInt();
        int letter=65;
        for(int i=1;i<=rows;i++){
            for (int j=1;j<=i;j++){
                System.out.print((char) (letter)+" ");
                letter+=1;
            }
            System.out.println();
        }
    }
}
```

Output:

```
Enter the number of rows
5
A
B C
D E F
G H I J
K L M N O
```


37. Write a program to print the below pattern based on the user entered a number of rows.

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Program:

```
import java.util.Scanner;
public class Pattern {
    public static void main(String args[]){
        Scanner obj= new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows=obj.nextInt();
        for(int i=1;i<=rows;i++){
            for(int j=1;j<=i;j++){
                System.out.print(j+" ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
Enter the number of rows
5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

38. Write a program to print the below pattern based on the user entered a number of rows.

```
1 1 1 1 1
2 2 2 2
3 3 3
4 4
5
```

Program:

```
import java.util.Scanner;
public class Pattern {
    public static void main(String args[]){
        int b=0;
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows=obj.nextInt();
        for(int i=rows;i>=1;i--){
            b+=1;
            for (int j=1;j<=i;j++){
                System.out.print(b+" ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
Enter the number of rows
5
1 1 1 1 1
2 2 2 2
3 3 3
4 4
5
```

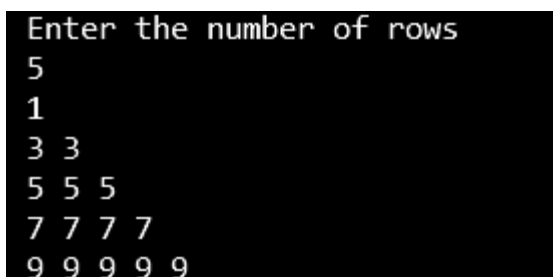
39. Write a program to print the below pattern based on the user entered a number of rows.

```
1
3 3
5 5 5
7 7 7 7
9 9 9 9 9
```

Program:

```
import java.util.Scanner;
public class Pattern{
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows=obj.nextInt();
        for(int i=1;i<=rows;i++){
            for(int j=1;j<=i;j++){
                System.out.print(i*2-1+" ");
            }
            System.out.println();
        }
    }
}
```

Output:

A screenshot of a terminal window with a black background and white text. It shows the prompt 'Enter the number of rows' followed by the input '5'. Below this, the program's output is displayed as a pattern of numbers: '1', '3 3', '5 5 5', '7 7 7 7', and '9 9 9 9 9' on separate lines.

```
Enter the number of rows
5
1
3 3
5 5 5
7 7 7 7
9 9 9 9 9
```

40. Write a program to print the below pattern based on the user entered a number of rows.

```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
```

Program:

```
import java.util.Scanner;
public class Pattern {
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows=obj.nextInt();
        for(int i=1;i<=rows;i++){
            for(int j=i;j>0;j--){
                System.out.print(j+" ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
Enter the number of rows
5
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
```

41. Write a program to print the below pattern based on the user's entered a number of rows.

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

Program:

```
import java.util.Scanner;
public class Pattern {
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows = obj.nextInt();
        for(int i=0;i<=rows+1;i++){
            for(int j=rows-i;j>0;j--){
                System.out.print(j+" ");
            }
            System.out.println();
        }
    }
}
```


Output:

```
Enter the number of rows
```

```
5
```

```
5 4 3 2 1
```

```
4 3 2 1
```

```
3 2 1
```

```
2 1
```

```
1
```

42. Write a program to print the below pattern based on the user entering a number of rows.

```
1
2  4
3  6  9
4  8  12  16
5  10 15  20  25
6  12 18  24  30  36
7  14 21  28  35  42  49
8  16 24  32  40  48  56  64
```

Program:

```
import java.util.Scanner;
public class Pattern {
    public static void main(String args[]){
        Scanner obj= new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows=obj.nextInt();
        for(int i=1;i<=rows;i++){
            for(int j=1;j<=i;j++){
                System.out.print(i*j+" ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
Enter thr number of rows
8
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
8 16 24 32 40 48 56 64
```

43. Write a program to print the below pattern based on the user entered a number of rows.

```
1
3 2
6 5 4
10 9 8 7
```

Program:

```
import java.util.Scanner;
public class Pattern{
    public static void main(String[] args) {
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the number of
rows");
        int rows=obj.nextInt();
        int x = 0, count = 0;
        for (int i = 1; i <= rows; i++) {
            x += i;
            count = x;
            for (int j = 1; j <= rows; j++) {
                if (j <= i) {
                    System.out.print(count + " ");
                    count -= 1;
                }
            }
            System.out.println("");
        }
    }
}
```

Output:

```
c:\java\jdk_7\bin>java com.gomarch2.class_000023
Enter the number of rows
5
1
3 2
6 5 4
10 9 8 7
15 14 13 12 11
```

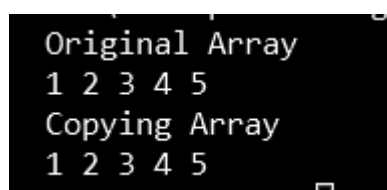
4. Basic programs in Arrays

44. Write a Program to copy all elements of one array into another array.

Program:

```
public class ArrayCopy {
    public static void main(String args[]){
        int [] a = new int [] {1, 2, 3, 4, 5};
        int b[] = new int[a.length];
        for(int i=0;i<a.length;i++)
        {
            b[i]=a[i];
        }
        System.out.println("Original Array");
        for(int i:a){
            System.out.print(i+" ");
        }
        System.out.println();
        System.out.println("Copying Array");
        for(int i:b){
            System.out.print(i+" ");
        }
    }
}
```

Output:

A screenshot of a terminal window showing the output of the Java program. The output consists of two lines of text: "Original Array" followed by "1 2 3 4 5" on the same line, and "Copying Array" followed by "1 2 3 4 5" on the next line. The text is white on a black background.

```
Original Array
1 2 3 4 5
Copying Array
1 2 3 4 5
```

45. Write a program to find the duplicate elements in an array.

Program:

```
public class DuplicateElements {  
    public static void main(String[] args) {  
        int [] a= new int [] {1, 2, 3, 4, 2, 7, 8, 8,  
3};  
        System.out.println("Duplicate elements in  
given array: ");  
        for(int i = 0; i < a.length; i++) {  
            for(int j = i + 1; j < a.length; j++) {  
                if(a[i] == a[j])  
                    System.out.println(a[j]);  
            }  
        }  
    }  
}
```

Output:

```
Duplicate elements in given array:
```

```
2
```

```
3
```

```
8
```

46. Write a program to print an element in reverse order.

Program:

```
public class ReverseArray {  
    public static void main(String[] args) {  
        int [] a = new int [] {1, 2, 3, 4, 5};  
        System.out.println("Original array: ");  
        for (int i = a.length-1; i >= 0; i--) {  
            System.out.print(a[i] + " ");  
        }  
    }  
}
```

Output:

```
Reverse array:  
5 4 3 2 1
```


47. Write a program to print the elements in the even position.

Program:

```
public class EvenPosition {  
    public static void main(String args[]){  
        int [] arr = new int [] {0, 1, 2, 3, 4, 5};  
        System.out.println("Elements of given array  
present on even position: ");  
        for (int i = 1; i < arr.length; i++) {  
            if(i%2==0){  
                System.out.println(arr[i]);  
            }  
        }  
    }  
}
```

Output:

```
Elements of given array present on even position:  
2  
4
```

48. Write a program to print the elements in the odd position.

Program:

```
public class OddPosition {  
    public static void main(String args[]){  
        int [] arr = new int [] {0, 1, 2, 3, 4, 5};  
        System.out.println("Elements of given array  
present on odd position: ");  
        for (int i = 1; i < arr.length; i++) {  
            if(arr[i]%2!=0){  
                System.out.println(arr[i]);  
            }  
        }  
    }  
}
```

Output:

```
Elements of given array present on odd position:  
3  
5
```

49. Write a program to find the largest value in an array.

Program:

```
public class LargestNumber{
    public static void main(String args[]){
        int a[]={33,67,90,20,10};
        int highest_number=0;
        for(int i=0;i<a.length;i++){
            if(a[i]>highest_number){
                highest_number=a[i];
            }
        }
        System.out.println("Largest Number
"+highest_number);
    }
}
```

Output:

```
Largest Number 90
```

50. Write a program to find the smallest value in the array.

Program:

```
public class LowestNumber {  
    public static void main(String args[]){  
        int a[]={15,20,30,8,70};  
        int lowest_number=a[0];  
        for(int i=0;i<a.length;i++){  
            if(a[i]<lowest_number){  
                lowest_number=a[i];  
            }  
        }  
        System.out.println("Lowest Number  
"+lowest_number);  
    }  
}
```

Output:

```
Lowest Number 8
```

51. Write a program to sum all the numbers in an array.

Program:

```
public class SumOfNumbers {  
    public static void main(String args[]){  
        int a[]={1,2,3,4,5};  
        int sum=0;  
        for(int i =0;i<a.length;i++){  
            sum+=a[i];  
        }  
        System.out.println("Sum of all the numbers in  
this array: "+sum);  
    }  
}
```

Output:

```
Sum of all the numbers in this array: 15
```

52. Write a program to sort the numbers in an array in ascending order.

Program:

```
public class AscendingOrder {
    public static void main(String args[])
    {
        int a[]={8,5,9,23,2,1};
        int temp=0;
        System.out.println("original Array");
        for(int i:a){
            System.out.print(i+" ");
        }
        for(int i=0;i<a.length;i++){
            for(int j=i+1;j<a.length;j++){
                if(a[i]>a[j]){
                    temp=a[i];
                    a[i]=a[j];
                    a[j]=temp;
                }
            }
        }
        System.out.println();
        System.out.println("Ascending order");
        for(int i:a){
            System.out.print(i+" ");
        }
    }
}
```

Output:

```
original Array  
8 5 9 23 2 1  
Ascending order  
1 2 5 8 9 23
```

53. Write a program to print the numbers in an array in descending order.

Program:

```
public class DescendingOrder {
    public static void main(String args[])
    {
        int a[]={8,5,9,23,2,1};
        int temp=0;
        System.out.println("original Array");
        for(int i:a){
            System.out.print(i+" ");
        }
        for(int i=0;i<a.length;i++){
            for(int j=i+1;j<a.length;j++){
                if(a[i]<a[j]){
                    temp=a[i];
                    a[i]=a[j];
                    a[j]=temp;
                }
            }
        }
        System.out.println();
        System.out.println("Descending order");
        for(int i:a){
            System.out.print(i+" ");
        }
    }
}
```


Output:

```
original Array  
8 5 9 23 2 1  
Descending order  
23 9 8 5 2 1
```

54. Write a program to find the third largest number in an array.

Program:

```
public class ThirdLargestNumber{
    public static void main(String args[])
    {
        int a[]={8,5,9,23,2,1};
        int temp=0;
        System.out.println("original Array");
        for(int i:a){
            System.out.print(i+" ");
        }
        for(int i=0;i<a.length;i++){
            for(int j=i+1;j<a.length;j++){
                if(a[i]<a[j]){
                    temp=a[i];
                    a[i]=a[j];
                    a[j]=temp;
                }
            }
        }
        System.out.println();
        System.out.println("Third largest number
"+a[2]);
    }
}
```

Output:

```
original Array
8 5 9 23 2 1
Third largest number 8
```

55. Write a program to find the second largest number in the array.

Program:

```
public class SecondLargestNumber{
    public static void main(String args[])
    {
        int a[]={8,5,9,23,2,1};
        int temp=0;
        System.out.println("original Array");
        for(int i:a){
            System.out.print(i+" ");
        }
        for(int i=0;i<a.length;i++){
            for(int j=i+1;j<a.length;j++){
                if(a[i]<a[j]){
                    temp=a[i];
                    a[i]=a[j];
                    a[j]=temp;
                }
            }
        }
        System.out.println();
        System.out.println("Second largest number
"+a[1]);
    }
}
```

Output:

```
original Array
8 5 9 23 2 1
Second largest number 9
```

56. Write a program to add two matrices.

Program:

```
public class AddMatrices {  
    public static void main(String args[]) {  
        int a[][] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7,  
8, 9 } };  
        int b[][] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7,  
8, 9 } };  
        int c[][] = new int[3][3];  
        System.out.println("Addition of two  
Matrices");  
        for (int i = 0; i < 3; i++) {  
            for (int j = 0; j < 3; j++) {  
                c[i][j] = a[i][j] + b[i][j];  
                System.out.print(c[i][j] + "\t");  
            }  
            System.out.println();  
        }  
    }  
}
```

Output:

```
Addition of two Matrices  
2      4      6  
8      10     12  
14     16     18
```

57. Write a program to subtract two matrices.

Program:

```
public class Subtraction {  
    public static void main(String args[]) {  
        int a[][] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7,  
8, 9 } };  
        int b[][] = { { 4, 5, 6 }, { 1, 2, 3 }, { 5,  
8, 2 } };  
        int c[][] = new int[3][3];  
        System.out.println("Subtraction of two  
Matrices");  
        for (int i = 0; i < 3; i++) {  
            for (int j = 0; j < 3; j++) {  
                c[i][j] = a[i][j] - b[i][j];  
                System.out.print(c[i][j] + "\t");  
            }  
            System.out.println();  
        }  
    }  
}
```

Output:

```
Subtraction of two Matrices  
-3    -3    -3  
3      3      3  
2      0      7
```

58. Write a program to product two matrices.

Program:

```
public class ProductMatrices {  
    public static void main(String args[]) {  
        int a[][] = { { 1, 1, 1 }, { 2, 2, 2 }, { 3,  
3, 3 } };  
        int b[][] = { { 1, 1, 1 }, { 2, 2, 2 }, { 3,  
3, 3 } };  
        int c[][] = new int[3][3];  
        System.out.println("Product of two  
Matrices");  
        for (int i = 0; i < 3; i++) {  
            for (int j = 0; j < 3; j++) {  
                c[i][j] = 0;  
                for (int k = 0; k < 3; k++) {  
                    c[i][j] += a[i][k] * b[k][j];  
                }  
                System.out.print(c[i][j] + "\t");  
            }  
            System.out.println();  
        }  
    }  
}
```

Output:

Product of two Matrices		
6	6	6
12	12	12
18	18	18

5. Function Example Program

59. write a program to swap two numbers.

Program:

```
public class Swap {  
    static void swapper(int a, int b) {  
        System.out.println("Before swapping a=" + a +  
" and b=" + b);  
        int temp;  
        temp = a;  
        a = b;  
        b = temp;  
        System.out.println("After swapping a=" + a +  
" and b=" + b);  
    }  
    public static void main(String args[]) {  
        swapper(20, 10);  
    }  
}
```

Output:

```
Before swapping a=20 and b=10  
After swapping a=10 and b=20
```

60. Write a program to swap two numbers without using a third variable.

Program:

```
public class SwapWithoutThird {  
    static void swapper(int a, int b) {  
        System.out.println("Before swapping a=" + a +  
" and b=" + b);  
        a = a + b;  
        b = a - b;  
        a = a - b;  
        System.out.println("After swapping a=" + a +  
" and b=" + b);  
    }  
  
    public static void main(String args[]) {  
        swapping(20, 10);  
    }  
}
```

Output:

```
Before swapping a=20 and b=10  
After swapping a=10 and b=20
```


61. Write a program to check whether the given number is positive or negative.

Program:

```
import java.util.Scanner;
public class PositiveOrNegative{

    static String positiveOrNegative(int a) {
        if (a > 0) {
            return "Positive";
        } else if (a < 0) {
            return "Negative";
        } else {
            return "Neither Positive nor Negative";
        }
    }

    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the number");
        int a = sc.nextInt();
        System.out.println("The entered number is a "
+ positiveOrNegative(a)+" number");
    }
}
```

Output:

```
enter the number  
-8  
firstnumber is Negative
```

```
enter the number  
2  
firstnumber is Positive
```

62. Write a program to find the factorial number using a recursive function.

Program:

```
import java.util.Scanner;

public class Recursion {
    static int factorial(int number) {
        if (number <= 1) {
            return 1;
        } else {
            return number * factorial(number - 1);
        }
    }

    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number");
        int number = sc.nextInt();
        System.out.println("Factorial of " + number +
            " is " + factorial(number));
    }
}
```

Output:

```
Enter the number
6
Factorial of 6 is 720
```

Types of functions:

- Without parameters and without return type
- Without parameters and with return type
- With parameters and without return type
- With parameters and with return type

63. Write a program without parameters and without return type.

Program:

```
public class TypesOfFunction {  
  
    static void add() {  
        int a = 15;  
        int b = 20;  
        int sum = a + b;  
        System.out.println("Addition of two numbers  
are:" + sum);  
    }  
  
    public static void main(String args[]) {  
        add();  
    }  
  
}
```

Output:

```
Addition of two numbers are:35
```

64. Write a program without parameters and with the return type.

Program:

```
public class TypesOfFunction {  
  
    static int add() {  
        int a = 15;  
        int b = 20;  
        int sum = a + b;  
        return sum;  
    }  
  
    public static void main(String args[]) {  
        System.out.println("Addition of two numbers  
are: " + add());  
    }  
}
```

Output:

```
Addition of two numbers are:35
```

65. Write a program with parameters and without return type.

Program:

```
public class TypesOfFunction {  
  
    static void add(int a, int b) {  
        int sum = a + b;  
        System.out.println("Addition of two numbers  
are: " + sum);  
    }  
  
    public static void main(String args[]) {  
        add(15, 20);  
    }  
}
```

Output:

```
Addition of two numbers are:35
```

66. Write a program with parameters and with the return type.

Program:

```
public class TypesOfFunction {  
  
    static int add(int a, int b) {  
        int sum = a + b;  
        return sum;  
    }  
  
    public static void main(String args[]) {  
        System.out.println("Addition of two numbers  
are: " + add(15, 20));  
    }  
}
```

Output:

```
Addition of two numbers are:35
```

6. Class and Object:

67. Write a program for classes and objects.

Program:

```
import java.util.Scanner;
class StudentResult {
    int id, eng, tam, mat, sci, soc, total;
    String name;
    float average;

    void getuserinput() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the student id");
        id = sc.nextInt();
        System.out.println("Enter the student name");
        name = sc.next();
        System.out.println("Enter the marks:");
        System.out.println("Enter english mark:");
        eng = sc.nextInt();
        System.out.println("Enter tamil marks:");
        tam = sc.nextInt();
        System.out.println("Enter math mark:");
        mat = sc.nextInt();
        System.out.println("Enter science mark:");
        sci = sc.nextInt();
        System.out.println("Enter social mark:");
        soc = sc.nextInt();
    }
}
```



```

void calculate() {
    total = eng + tam + mat + sci + soc;
    average = total / 5;
}

void display() {
    System.out.println("Student Information");
    System.out.println("-----
    ---");
    System.out.println("Student Id: " + id);
    System.out.println("Student Name: " + name);
    System.out.println("English: " + eng);
    System.out.println("Tamil: " + tam);
    System.out.println("Maths: " + mat);
    System.out.println("Science: " + sci);
    System.out.println("Social: " + soc);
    System.out.println("Total: " + total);
    System.out.println("Average: " + average);

}

}

class Test {
    public static void main(String args[]) {
        StudentResult s1 = new StudentResult();
        s1.getUserinput();
        s1.calculate();
        s1.display();
    }
}

```

Output:

```
Enter the student id
111
Enter the student name
vicky
Enter the marks:
Enter english mark:
98
Enter tamil marks:
68
Enter math mark:
78
Enter science mark:
98
Enter social mark:
89
Student Information
-----
Student Id: 111
Student Name: vicky
English: 98
Tamil: 68
Maths: 78
Science: 98
Social: 89
Total: 431
Average: 86.0
PS D:\gomathi_class> 
```

7. Constructor:

Types of Constructor:

- Default constructor
- Parameterized constructor

68. Example for Default Constructor

Program:

```
class DefaultConstructor {  
    DefaultConstructor() {  
        System.out.println("Default Constructor is  
created");  
    }  
    public static void main(String args[]) {  
        DefaultConstructor dc = new  
DefaultConstructor();  
    }  
}
```

Output:

```
Default Constructor is created
```

69. Example for parameterized constructor

Program:

```
public class ParameterizedConstructor {  
    int id;  
    String name;  
  
    ParameterizedConstructor(int n, String name1) {  
        id = n;  
        name = name1;  
    }  
  
    void display() {  
        System.out.println(id + " " + name);  
    }  
  
    public static void main(String args[]) {  
        ParameterizedConstructor pc = new  
ParameterizedConstructor(1, "Abbi");  
        pc.display();  
    }  
}
```

Output:

```
1 Abbi
```

70. Example for constructor overloading.

Program:

```
public class ParameterizedConstructor {
    int id;
    String name;
    int age;
    ParameterizedConstructor(int n,String name1) {
        id = n;
        name = name1;
    }
    ParameterizedConstructor(int n,String name1,int
a) {
        id = n;
        name = name1;
        age = a;
    }
    void display() {
        System.out.println(id + " " + name + " " +
age);
    }
    public static void main(String args[]) {
        ParameterizedConstructor pc = new
ParameterizedConstructor(1, "Abbi");
        ParameterizedConstructor pc1 = new
ParameterizedConstructor(2, "annu", 20);
        pc.display();
        pc1.display();
    }
}
```

Output:

```
1 Abbi 0
2 annu 20
```

8. Inheritance:

Types of Inheritance:

- Single Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Multiple Inheritance(does not support)
- Hybrid Inheritance

71. Write a program for single inheritance.

Program:

```
class Animal {
    void eat() {
        System.out.println("Eating...");
    }

    void sleep() {
        System.out.println("Sleeping...");
    }
}

class Lion extends Animal {
    void roar() {
        System.out.println("Roaring...");
    }
}

class SingleInheritance{
```

```
public static void main(String args[]) {  
    Lion l = new Lion();  
    l.eat();  
    l.sleep();  
    l.roar();  
}  
}
```

Output:

```
Eating...  
Sleeping...  
Roaring...
```


72. Example program for multilevel inheritance.

Program:

```
class Animal {
    void eat() {
        System.out.println("Eating...");
    }

    void sleep() {
        System.out.println("Sleeping...");
    }
}

class Lion extends Animal {
    void roar() {
        System.out.println("Roaring...");
    }
}

class SmallLion extends Lion {
    void play() {
        System.out.println("Playing...");
    }
}

class MultiLevelInheritance {
    public static void main(String args[]) {
        SmallLion sm = new SmallLion();
        sm.eat();
        sm.sleep();
        sm.roar();
    }
}
```

```
        sm.play();  
    }  
}
```

Output:

```
Eating...  
Sleeping...  
Roaring...  
Playing...
```

73. Example for Hierarchical Inheritance.

Program:

```
class Animal {
    void eat() {
        System.out.println("Eating...");
    }

    void sleep() {
        System.out.println("Sleeping...");
    }
}

class Lion extends Animal {
    void roar() {
        System.out.println("Roaring...");
    }
}

class Cat extends Animal {
    void meow() {
        System.out.println("Meowww...");
    }
}

class HierarchicalInheritance {
    public static void main(String args[]) {
        Lion l = new Lion();
        l.eat();
        l.sleep();
        l.roar();
    }
}
```

```
Cat c=new Cat();  
c.eat();  
c.sleep();  
c.meow();  
}  
}
```

Output:

A screenshot of a terminal window with a black background and white text. The output consists of six lines: "Eating...", "Sleeping...", "Roaring...", "Eating...", "Sleeping...", and "Meowww...". The text is left-aligned and appears to be the result of the program's execution.

```
Eating...  
Sleeping...  
Roaring...  
Eating...  
Sleeping...  
Meowww...
```

9. Polymorphism:

74. Write a program for method overloading changing the number of arguments.

Program:

```
class MethodOverloading {
    static int add(int a, int b){
        return a + b;
    }

    static int add(int a, int b, int c){
        return a + b + c;
    }
}

class Test{
    public static void main(String[] args) {
        System.out.println(Adder.add(11, 11));
        System.out.println(Adder.add(11, 11, 11));
    }
}
```

Output:

```
22
33
```

75. Write a program for method overloading by changing the datatype of the parameters.

Program:

```
class MethodOverloading {
    static int add(int a, int b) {
        return a + b;
    }

    static double add(double a, double b) {
        return a + b;
    }
}

class Test{
    public static void main(String[] args) {
        System.out.println(MethodOverloading.add(11, 11));
        System.out.println(MethodOverloading.add(11.2d,
11.2d));
    }
}
```

Output:

```
22
22.4
```

76. Write a program for method overriding.

Program:

```
class Course {
    int getCourseDuration() {
        return 0;
    }
}

class Java extends Course {
    @Override
    int getCourseDuration() {
        return 90;
    }
}

class MySql extends Course {
    @Override
    int getCourseDuration() {
        return 30;
    }
}

public class MethodOverriding {
    public static void main(String args[]) {
        Java j = new Java();
        MySql m = new MySql();
        System.out.println("Java course duration is:"
+ j.getCourseDuration() + " days.");
        System.out.println("Mysql course duration
is:" + m.getCourseDuration() + " days.");
    }
}
```

```
}
```

Output:

```
Java course duration is:90 days.  
Mysql course duration is:30 days.
```


77. Write a program the super keyword to refer to an immediate parent class instance variable.

Program:

```
class OceanAcademy {
    String courseName = "Not Available";
}

class Student extends OceanAcademy {
    String courseName = "Java";

    void displayCourseName() {
        System.out.println("Student class
courseName:" + courseName);
        System.out.println("OceanAcademy class
courseName:" + super.courseName);
    }
}

public class Test {
    public static void main(String args[]) {
        Student s = new Student();
        s.displayCourseName();
    }
}
```

Output:

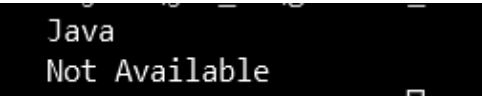
```
Student class courseName:Java
OceanAcademy class courseName:Not Available
```

78. Write a program with the super keyword to invoke parent class methods.

Program:

```
class OceanAcademy {
    String getCourseDetails() {
        return "Not Available";
    }
}
class Student extends OceanAcademy {
    @Override
    String getCourseDetails() {
        return "Java";
    }
    void display() {
        System.out.println(getCourseDetails());
        System.out.println(super.getCourseDetails());
    }
}
public class Test {
    public static void main(String args[]) {
        Student s = new Student();
        s.display();
    }
}
```

Output:

A screenshot of a terminal window showing the output of the Java program. The output consists of two lines: "Java" on the first line and "Not Available" on the second line. The text is white on a black background.

```
Java
Not Available
```

79. Write a program using the super constructor.

Program:

```
class OceanAcademy {
    int id;
    String courseName;
    OceanAcademy(int id, String courseName) {
        this.id = id;
        this.courseName = courseName;
    }
}

class Student extends OceanAcademy {
    String studName;
    Long mobileNo;
    Student(int id, String courseName, String
studName, long mobileNo) {
        super(id, courseName);
        this.studName = studName;
        this.mobileNo = mobileNo;
    }

    public void display() {
        System.out.println("Student details:id=" + id
+ " Student Name=" + studName + " Course=" +
courseName + " Mobile Number=" + mobileNo);
    }
}

public class Test {
    public static void main(String args[]) {
        Student s = new Student(123, "Java", "Arun",
9988776655L);
    }
}
```

```
        s.display();  
    }  
  
}
```

Output:


```
Student details:id=123 Student Name=Arun Course=Java Mobile Number=9988776655
```

80. Write a program to implement an interface.

Program:

```
interface InterfaceOne{
    void display();
}
interface InterfaceTwo{
    void show();
}
class Test implements InterfaceOne,InterfaceTwo {
    @Override
    public void show() {
        System.out.println("Ocean Academy");
    }
    @Override
    public void display() {
        System.out.println("Welcome");
    }
    public static void main(String args[]){
        Test t=new Test();
        t.display();
        t.show();
    }
}
```

Output:

A screenshot of a terminal window showing the output of the Java program. The output consists of two lines: "Welcome" on the first line and "Ocean Academy" on the second line. The text is white on a black background.

```
Welcome
Ocean Academy
```

10. Exception Handling.

81. Write a simple program for exception handling.

Program:

```
public class ExceptionExample {  
    public static void main(String args[]){  
        try{  
            int a =100/0;  
        }catch (ArithmeticException e){  
            System.out.println(e);  
        }  
        System.out.println("remaining part of  
execution");  
    }  
}
```

Output:

```
java.lang.ArithmeticException: / by zero  
finally block always executed  
remaining part executte
```

82. Write a program for exception handling using the throw keyword.

Program:

```
import java.util.Scanner;

public class ExceptionExample {
    public static void positive(int a){
        if(a<0){
            throw new ArithmeticException("the value
is negative");
        }
        else{
            System.out.println("the value is
positive");
        }
    }
    public static void main(String args[]){
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the number");
        int a=obj.nextInt();
        positive(a);
        System.out.println("remaining code");
    }
}
```

Output:

```
Enter the number  
5  
the value is positive  
remaining code
```

```
Enter the number  
-5  
Exception in thread "main" java.lang.ArithmeticException: the value is negative  
    at ExceptionExample.positive(ExceptionExample.java:6)  
    at ExceptionExample.main(ExceptionExample.java:17)
```


11. String:

83. Write a program for the String methods.

Program:

```
public class ExampleString {
    public static void main(String args[]) {
        String name = "Ocean Academy";
        char ch = name.charAt(4);
        // it returns the char value at the 4th index
        System.out.println(ch);
        // it concatenate string with the given
string
        name = name.concat("is software training
institute");
        System.out.println(name);
        // if String is present it returns true
        System.out.println(name.contains("ocean"));
        // it returns true the string ends with the
given string
        System.out.println(name.endsWith("institute"));
        System.out.println(String.format("name is %s
", name));
        //it returns the start index value for the
given string
        System.out.println(name.indexOf("is"));
        //it returns true if the string is empty
        System.out.println(name.isEmpty());
        //it returns the string after joining "-"
        String str = String.join("-", "welcome",
"to", "ocean");
```

```

        System.out.println(str);
        //it returns last index of "e"
        System.out.println(str.lastIndexOf("e"));
        //it returns the length of the string
        System.out.println(str.length());
        //it replace "ocean" with "Ocean"
        String replace = str.replace("ocean",
"Ocean");
        System.out.println(replace);
        //it replace all the "o" with "O"
        String replaceall = str.replaceAll("o", "O");
        System.out.println(replaceall);
    }
}

```

Output:

```

n
Ocean Academyis software training institute
false
true
name is Ocean Academyis software training institute
13
false
welcome-to-ocean
13
16
welcome-to-Ocean
welcOme-tO-Ocean

```

84. Given a string name, e.g. "Bob", return a greeting of the form "Hello Bob!".

helloName("Ocean") → "Hello Ocean!"

helloName("Alice") → "Hello Alice!"

helloName("X") → "Hello X!"

Program:

```
public class HelloName{

    public static String helloName(String name) {
        return "Hello "+name+"!";
    }

    public static void main(String[] args) {

        System.out.println(helloName("Ocean"));

    }
}
```

Output:

A screenshot of a terminal window with a black background. The first line shows the output "Hello Ocean!". The second line shows the command prompt "PS E:\gomathi\Java basic program>".

```
Hello Ocean!
PS E:\gomathi\Java basic program>
```

85. Given a string, return a new string made of 3 copies of the last 2 chars of the original string. The string length will be at least 2.

`extraEnd("Hello") → "lololo"`

`extraEnd("ab") → "ababab"`

`extraEnd("Hi") → "HiHiHi"`

Program:

```
public class LastTwoLetter{

    public static String LastTwo(String name) {
        String str=name.substring(name.length()-2,
name.length());
        return str+str+str;

    }

    public static void main(String[] args) {

        System.out.println(LastTwo("Ocean"));

    }

}
```

Output:



```
anahan
PS E:\gomathi\Java basic program>
```

86. Given a string, return the string made of its first two chars, so the String "Hello" yields "He". If the string is shorter than length 2, return whatever there is, so "X" yields "X", and the empty string "" yields the empty string "". Note that `str.length()` returns the length of a string.

```
firstTwo("Hello") → "He"
firstTwo("ABCDEFGH") → "AB"
firstTwo("AB") → "AB"
```

Program:

```
public class FirstTwo {
    public static String firstTwo(String name) {
        if(name.length()<2){
            return name;
        }
        else
        {
            return name.substring(0, 2);
        }
    }

    public static void main(String[] args) {
        System.out.println(firstTwo("Ocean"));
    }
}
```

Output:

```
0c
PS E:\gomathi\Java basic program>
```

87. Write a program to reverse words.

Input: "ocean Academy"

output: " Academy ocean"

Program:

```
public class ReverseWords {

    static String reverseWords(String name) {
        String str[] = name.split(" ");
        String answer = " ";

        for(int i = str.length - 1; i >= 0; i--) {
            answer += str[i] + " ";
        }

        String
reversedString = answer.substring(0, answer.length());
        return reversedString;

    }

    public static void main(String[] args) {

        System.out.println(reverseWords("Ocean
Academy"));

    }

}
```

Output:

The image shows a terminal window with a black background. The text 'Academy Ocean' is displayed in a light blue or cyan monospaced font. The word 'Academy' is followed by a space and then the word 'Ocean'.

12. Array List.

88. Write a program for sorting an array list in ascending order.

Program:

```
import java.util.ArrayList;

public class SortingArray {
    public static void main(String args[]){
        ArrayList<Integer> a=new
ArrayList<Integer>();
        a.add(20);
        a.add(10);
        a.add(60);
        a.add(50);
        a.add(40);
        System.out.println("Before sorting
element"+a);
        int temp=0;
        for(int i=0;i<a.size();i++){
            for(int j=i+1;j<a.size();j++){
                if(a.get(i)>a.get(j)){
                    temp=a.get(i);
                    a.set(i,a.get(j));
                    a.set(j,temp);
                }
            }
        }
    }
}
```

```
        System.out.println("After sorting  
element"+a) ;  
    }  
  
}
```

Output:

```
Before sorting element[20, 10, 60, 50, 40]  
After sorting element[10, 20, 40, 50, 60]  
PS E:\gomathi\Java basic program> 
```


89. Write a program for sorting an array list in descending order.

Program:

```
import java.util.ArrayList;
public class SortingArray {
    public static void main(String args[]){
        ArrayList<Integer> a=new
ArrayList<Integer>();
        a.add(20);
        a.add(10);
        a.add(60);
        a.add(50);
        a.add(40);
        System.out.println("Before sorting
element"+a);
        int temp=0;
        for(int i=0;i<a.size();i++){
            for(int j=i+1;j<a.size();j++){
                if(a.get(i)<a.get(j)){
                    temp=a.get(i);
                    a.set(i,a.get(j));
                    a.set(j,temp);
                }
            }
        }
        System.out.println("After sorting
element"+a);
    }
}
```

Output:

```
Before sorting element[20, 10, 60, 50, 40]  
After sorting element[60, 50, 40, 20, 10]
```

90. Write a program for reversing an ArrayList.

Program:

```
import java.util.ArrayList;

public class ReverseArray {
    public static void main(String args[]){
        ArrayList<Integer> a=new
ArrayList<Integer>();
        ArrayList<Integer> b=new
ArrayList<Integer>();
        a.add(1);
        a.add(2);
        a.add(3);
        a.add(4);
        a.add(5);
        System.out.println("Before a reverse "+a);
        for(int i=a.size()-1;i>=0;i--){
            b.add(a.get(i));
        }
        System.out.println("After a reverse"+b);
    }
}
```

Output:

```
Before a reverse [1, 2, 3, 4, 5]
After a reverse[5, 4, 3, 2, 1]
```

91. Find the largest and smallest element in the array list.

Program:

```
public class LargestNumber{
    public static void main(String args[]){
        ArrayList<Integer> a=new
ArrayList<Integer>();
        a.add(10);
        a.add(50);
        a.add(20);
        a.add(70);
        a.add(90);
        System.out.println("Before sorting"+a);
        int temp=0;
        for(int i =0;i<a.size();i++){
            for(int j=i+1;j<a.size();j++){
                if(a.get(i)<a.get(j)){
                    temp=a.get(i);
                    a.set(i,a.get(j));
                    a.set(j,temp);

                }
            }

        }

        System.out.println("After sorting"+a);
        System.out.println("greatest number"
+a.get(0));
```

```
        System.out.println("Smallest  
number"+a.get(a.size()-1));  
  
    }  
}
```

Output:

```
Before sorting[10, 50, 20, 70, 90]  
After sorting[90, 70, 50, 20, 10]  
greatest number90  
Smallest number10
```

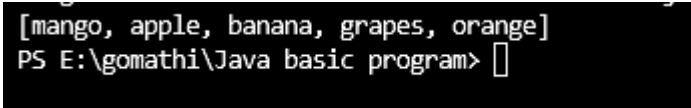
13. Linked list.

92. Write a program to add elements into a LinkedList in java.

Program:

```
public class LinkedListExample {  
    public static void main(String args[]){  
        LinkedList<String> fruits=new  
LinkedList<String>();  
        fruits.add("mango");  
        fruits.add("apple");  
        fruits.add("banana");  
        fruits.add("grapes");  
        fruits.add("orange");  
        System.out.println(fruits);  
    }  
}
```

Output:



```
[mango, apple, banana, grapes, orange]  
PS E:\gomathi\Java basic program>
```

93. Write a program to add the elements in the first and last in the LinkedList.

Program:

```
public class LinkedListExample {
    public static void main(String args[]){
        LinkedList<String> fruits=new
LinkedList<String>();

        fruits.add("mango");
        fruits.add("apple");
        fruits.add("banana");
        fruits.add("grapes");
        fruits.add("orange");
        System.out.println("Before adding element in
the list"+fruits);
        fruits.addFirst("cherry");
        System.out.println("After adding element in
the list"+fruits);
        fruits.addLast("guva");
        System.out.println("after adding element in
the last"+fruits);
    }
}
```

Output:

```
Before adding element in the list[mango, apple, banana, grapes, orange]
After adding element in the list[cherry, mango, apple, banana, grapes, orange]
after adding element in the last[cherry, mango, apple, banana, grapes, orange, guva]
PS E:\gomathi\Java basic program> █
```

94. Write a program to remove the first and last element in the LinkedList.

Program:

```
import java.util.LinkedList;
public class LinkedListExample {
    public static void main(String args[]){
        LinkedList<String> fruits=new
LinkedList<String>();
        fruits.add("mango");
        fruits.add("apple");
        fruits.add("banana");
        fruits.add("grapes");
        fruits.add("orange");
        System.out.println("Before removing element
in the list"+fruits);
        fruits.removeFirst();
        System.out.println("After removing element in
the first"+fruits);
        fruits.removeLast();
        System.out.println("After removing element in
the last"+fruits);
    }
}
```

Output:

```
Before removing element in the list[mango, apple, banana, grapes, orange]
After removing element in the first[apple, banana, grapes, orange]
after removing element in the last[apple, banana, grapes]
PS E:\gomathi\Java basic program>
```


95. Write a program to get the first and last element in the LinkedList.

Program:

```
import java.util.LinkedList;

public class LinkedListExample {
    public static void main(String args[]){
        LinkedList<String> fruits=new
LinkedList<String>();
        fruits.add("mango");
        fruits.add("apple");
        fruits.add("banana");
        fruits.add("grapes");
        fruits.add("orange");
        System.out.println("Element in the list
"+fruits);
        String a=fruits.getFirst();
        System.out.println("Getting an element in the
first "+a);
        String b=fruits.getLast();
        System.out.println("Getting an element in the
last "+b);
    }
}
```

Output:

```
element in the list[mango, apple, banana, grapes, orange]
getting element in the first mango
getting element in the last orange
PS E:\gomathi\Java basic program> █
```

14. Hash Map.

96. Write a program for adding elements in the Hashmap.

Program:

```
import java.util.HashMap;

public class HashMapExample {
    public static void main(String args[]){
        HashMap<Integer, String> oceanCourse=new
HashMap<Integer, String>();
        oceanCourse.put(1,"C");
        oceanCourse.put(2, "C++");
        oceanCourse.put(3,"java");
        oceanCourse.put(4, "Python");
        oceanCourse.put(5,"javascript");
        System.out.println(oceanCourse);

    }

}
```

Output:

```
{1=C, 2=C++, 3=java, 4=Python, 5=javascript}
PS E:\gomathi\Java basic program>
```

97. Write a program to get an element, remove an element, and clear all elements in the Hashmap.

Program:

```
import java.util.HashMap;
public class HashMapExample {
    public static void main(String args[]){
        HashMap<Integer, String> oceanCourse=new
HashMap<Integer, String>();
        oceanCourse.put(1, "C");
        oceanCourse.put(2, "C++");
        oceanCourse.put(3, "java");
        oceanCourse.put(4, "Python");
        oceanCourse.put(5, "javascript");
        System.out.println(oceanCourse);
        System.out.println("Getting an element in the
HashMap "+oceanCourse.get(4));
        System.out.println("Removing an element in
the HashMap "+oceanCourse.remove(5));
        System.out.println(oceanCourse);
        System.out.println("Clearing all the element in the
HashMap ");
        oceanCourse.clear();
        System.out.println(oceanCourse);
    }
}
```

Output:

```
{1=C, 2=C++, 3=java, 4=Python, 5=javascript}  
Getting a element in the map Python  
Remove a element in the map javascript  
{1=C, 2=C++, 3=java, 4=Python}  
{}  
PS E:\gomathi\Java basic program> |
```

98. Write a program to get elements through looping in a hash map.

Program:

```
import java.util.HashMap;
public class HashMapExample {
    public static void main(String args[]){
        HashMap<Integer, String> oceanCourse=new
HashMap<Integer, String>();
        oceanCourse.put(1, "C");
        oceanCourse.put(2, "C++");
        oceanCourse.put(3, "java");
        oceanCourse.put(4, "Python");
        oceanCourse.put(5, "javascript");
        for(Integer i: oceanCourse.keySet()){
            System.out.println(i);
        }
        for(String i: oceanCourse.values()){
            System.out.println(i);
        }
        for(Integer i : oceanCourse.keySet()){
            System.out.println("key: "+i+"value:
"+oceanCourse.get(i));
        }
    }
}
```

Output:

```
{1=C, 2=C++, 3=java, 4=Python, 5=javascript}
Getting a element in the map Python
1
2
3
4
5
C
C++
java
Python
javascript
key: 1value: C
key: 2value: C++
key: 3value: java
key: 4value: Python
key: 5value: javascript
PS E:\gomathi\Java basic program> █
```

15. Hash set

99. Write a program to add items, check for an item, remove an item and remove all the items in an Hashset.

Program:

```
import java.util.HashSet;
public class HashSetExample {
    public static void main(String args[]){
        HashSet<String> course=new HashSet<String>();
        course.add("C++");
        course.add("java");
        course.add("python");
        course.add("javascript");
        course.add("SQL");
        System.out.println(course);
        course.contains("java");//check an item
exists in a hashset
        course.remove("python");//remove an item in
hashset
        System.out.println(course);
        course.clear();//remove all item in a hashset
        System.out.println(course);
    }
}
```

Output:

```
[C++, python, java, javascript, SQL]
[C++, java, javascript, SQL]
[]
PS E:\gomathi\Java basic program> []
```

16. LinkedHashSet

100. Write a program to add items into a linked Hashset.

Program:

```
import java.util.LinkedHashSet;

public class LinkedHashSetExample {
    public static void main(String args[]){
        LinkedHashSet<String> ocean=new
LinkedHashSet<String>();
        ocean.add("C++");
        ocean.add("Java");
        ocean.add("Javascript");
        ocean.add("Python");
        ocean.add("SQL");
        System.out.println(ocean);

    }

}
```

Output:

```
[C++, Java, Javascript, Python, SQL]
```


17.IO Package:

101. Write a program to write data into a file using OutputStream.

Program:

```
import java.io.OutputStream;
import java.io.Writer;
import java.io.FileOutputStream;
import java.io.OutputStreamWriter;

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

        try {

            OutputStream outputStream = new
FileOutputStream("D:\\input.txt");
            Writer outputStreamWriter = new
OutputStreamWriter(outputStream);

            outputStreamWriter.write("Hello Ocean");
            System.out.println("data writed
successfully");

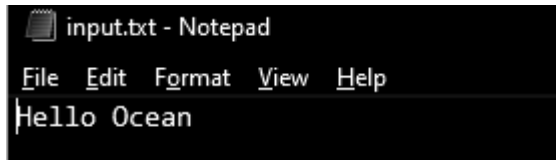
            outputStreamWriter.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
}
```

Output:

```
data writed successfully
```

input.txt:



input.txt - Notepad

File Edit Format View Help

Hello Ocean

102. Write a program to read data from a file using InputStream.

Program:

```
import java.io.InputStream;
import java.util.Scanner;
import java.io.FileInputStream;
public class IoExampleProgram {
    public static void main(String args[]) {
        try {

            InputStream input = new
FileInputStream("D:\\input.txt");
            Scanner myReader = new Scanner(input);
            System.out.println("Data read from the file: ");
            while (myReader.hasNextLine()) {
                String data = myReader.nextLine();
                System.out.println(data);
            }

            // Close the input stream
            myReader.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output:

```
Data read from the file:
welcome to ocean academy
```

103. Write a program to write data into a file using `FileOutputStream`.

Program:

```
import java.io.FileOutputStream;
public class IoExampleProgram {
    public static void main(String args[]){
        try{
            FileOutputStream fopen=new
FileOutputStream("D:\\input.txt");
            String str="Welcome to Ocean.";
            byte b[]=str.getBytes();//converting
string into byte array
            fopen.write(b);
            fopen.close();
            System.out.println("success...");
        }catch(Exception e){
            System.out.println(e);
        }
    }
}
```

Output:

A terminal window with a black background and white text. The text "success..." is displayed on a single line.

input.txt

A terminal window with a black background and white text. The first line of the file "input.txt" is shown, starting with a line number "1" followed by the text "Welcome to Ocean.".

104. Write a program to read data from a file using FileInputStream.

Program:

```
import java.io.FileInputStream;
public class IoExampleProgram {
    public static void main(String args[]){
        try{
            FileInputStream fopen=new
FileInputStream("D:\\input.txt");
            int i=0;
            while((i=fopen.read())!=-1){
                System.out.print((char)i);
            }
            fopen.close();
        }catch(Exception e){System.out.println(e);}
    }
}
```

Output:

```
Welcome to Ocean.
PS D:\gomathi\java> 
```

105. Write a program to write data into a file using BufferedOutputStream.

Program:

```
import java.io.FileOutputStream;
public class IoExampleProgram {
    public static void main(String args[]){
        try{
            FileOutputStream fopen=new
FileOutputStream("D:\\input.txt");
            BufferedOutputStream bos=new
BufferedOutputStream(fopen);
            String str="Welcome to Ocean Academy.";
            byte b[]=str.getBytes();//converting
string into byte array
            bos.write(b);
            bos.flush();
            bos.close();
            fopen.close();
            System.out.println("successfully
writed...");
        }catch(Exceptione){System.out.println(e);
        }
    }
}
```

Output:

```
successfully writed...  
PS D:\gomathi\java> 
```

Input.txt

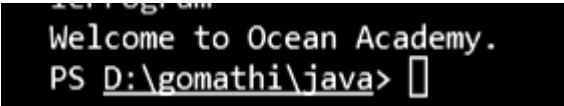
```
Welcome to Ocean Academy.
```

106. Write a program to read data from a file using `BufferedInputStream`.

Program:

```
import java.io.FileInputStream;
import java.io.BufferedInputStream;
public class IoExampleProgram{
    public static void main(String args[]){
        try{
            FileInputStream fis=new
FileInputStream("D:\\input.txt");
            BufferedInputStream bis=new
BufferedInputStream(fis);
            int i;
            while((i=bis.read())!=-1){
                System.out.print((char)i);
            }
            bis.close();
            fis.close();
        }catch(Exception e){System.out.println(e);}
    }
}
```

Output:



```
Welcome to Ocean Academy.
PS D:\gomathi\java>
```


107. Write a program for ByteArrayOutputStream.

Program:

```
import java.io.*;
public class IoExampleProgram{
    public static void main(String args[]) throws
Exception{
    byte[] a={65,66,67,68};
    FileOutputStream fout1=new
FileOutputStream("D:\\input1.txt");
    FileOutputStream fout2=new
FileOutputStream("D:\\input2.txt");
    ByteArrayOutputStream bout=new
ByteArrayOutputStream();
    bout.write(a);
    bout.writeTo(fout1);
    bout.writeTo(fout2);
    bout.flush();
    bout.close();//has no effect
    System.out.println("Success...");
}
}
```

Output:

```
Success...
PS D:\gomathi\java> █
```

Input1.txt

ABCD

Input2.txt

ABCD

108. Write a program for ByteArrayInputStream.

Program:

```
import java.io.*;
public class IoExampleProgram{
    public static void main(String[] args) throws
IOException {
        byte[] b = { 35, 36, 37, 38 };
        // Create the new byte array input stream
        ByteArrayInputStream bais = new
ByteArrayInputStream(b);
        int k = 0;
        while ((k = bais.read()) != -1) {
            //Conversion of a byte into character
            char ch = (char) k;
            System.out.println("ASCII value of Character
is:" + k + "; Special character is: " + ch);
        }
    }
}
```

Output:

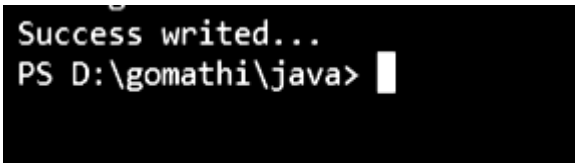
```
ASCII value of Character is:35; Special character is: #
ASCII value of Character is:36; Special character is: $
ASCII value of Character is:37; Special character is: %
ASCII value of Character is:38; Special character is: &
```

109. Write a program for FilterOutputStream.

Program:

```
import java.io.*;
public class IoExampleProgram{
    public static void main(String args[]) throws
IOException{
        File data = new
File("D:/gomathi/java/Array/FunctionsProgram/src/inpu
t.txt ");
        FileOutputStream file = new
FileOutputStream(data);
        FilterOutputStream filter = new
FilterOutputStream(file);
        String s="Welcome to Ocean Academy.";
        byte b[]=s.getBytes();
        filter.write(b);
        filter.flush();
        filter.close();
        file.close();
        System.out.println("Success writed...");
    }
}
```

Output:



```
Success writed...
PS D:\gomathi\java>
```

Input.txt

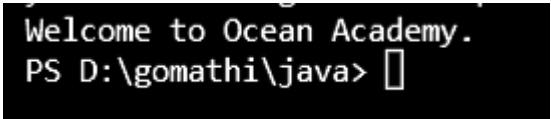
```
Welcome to Ocean Academy.
```

110. Write a program for FilterInputstream.

Program:

```
import java.io.*;
public class IoExampleProgram{
    public static void main(String args[]) throws
IOException{
        File data = new
File("D:/gomathi/java/Array/FunctionsProgram/src/inpu
t.txt ");
        FileInputStream file = new
FileInputStream(data);
        FilterInputStream filter = new
BufferedInputStream(file);
        int k =0;
        while((k=filter.read())!=-1){
            System.out.print((char)k);
        }
        filter.close();
        file.close();
    }
}
```

Output:



Welcome to Ocean Academy.
PS D:\gomathi\java> □

18. JDBC Connectivity:

111. Write a program for reading the data from the database.

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class DatabaseProgram {
    public static void main(String[] args) throws
Exception {
    try {
        //driver class

Class.forName("com.mysql.cj.jdbc.Driver");
        //creating a connection
        Connection con
=DriverManager.getConnection("jdbc:mysql://localhost:
3306/demo_class", "root", "root");
        System.out.println("Connected database
successfully.....");
        // creating a statement
        Statement stmt = con.createStatement();
        //execute the query
        System.out.println("read the data from
database");
        String query = "select * from student";
        ResultSet rs = stmt.executeQuery(query);
        while (rs.next()) {
```

```

        for (int i = 1; i <= 4; i++) {
            System.out.print(rs.getString(i)
+ " ");

        }
        System.out.println("");
    }
    //close the connection
    stmt.close();
    rs.close();

} catch (Exception ex) {
    System.out.println(ex);
}
}
}

```

Output:

```

Connected database successfully.....
read the data from database
1 aaaa 1234567890 ocean academy
2 bbbb 1234569801
3 aaaa 1234567891 ocean academy

```


112. Write a program for inserting the data into the table.

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class DatabaseProgram {
    public static void main(String[] args) throws
Exception {
    try {
        //driver class

Class.forName("com.mysql.cj.jdbc.Driver");
        //creating a connection
        Connection con
=DriverManager.getConnection("jdbc:mysql://localhost:
3306/demo_class", "root", "root");
        System.out.println("Connected database
successfully.....");
        // creating a statement
        Statement stmt = con.createStatement();
        //execute the query
        //System.out.println("read the data from
database");
        String query = "insert into student(id,
name, mobile) values (4,'ddd','7448776015)";
        stmt.executeUpdate(query);
        //close the connection
```

```
        System.out.println("data inserted  
successfully");  
        stmt.close();  
    } catch (Exception ex) {  
        System.out.println(ex);  
    }  
}  
}
```

Output:

```
Connected database successfully.....  
data inserted successfully
```

113. Write a program to create a database in SQL.

Program:

```
//creating a database:
public class DatabaseProgram {
    static final String DB_URL =
"jdbc:mysql://localhost/";
    static final String USER = "root";
    static final String PASS = "root";

    public static void main(String[] args) {
        // Open a connection
        try (Connection conn =
DriverManager.getConnection(DB_URL, USER, PASS);
            Statement stmt =
conn.createStatement();) {
            String query = "CREATE DATABASE
STUDENTS";
            stmt.executeUpdate(query);
            System.out.println("Database created
successfully...");
        } catch (SQLException e) {
            System.out.println(e);
        }
    }
}
```

Output:

```
Database created successfully...
```

114. Write a program to create a table in SQL.

Program:

```
public class DatabaseProgram {
    static final String DB_URL =
"jdbc:mysql://localhost/students";
    static final String USER = "root";
    static final String PASS = "root";

    public static void main(String[] args) {
        // Open a connection
        try (Connection conn =
DriverManager.getConnection(DB_URL, USER, PASS);
            Statement stmt =
conn.createStatement();) {
            String query = "create table
studentInformation(id int not null,name
varchar(255),age int, primary key(id))";
            stmt.executeUpdate(query);
            System.out.println("created table
successfully...");
        } catch (SQLException e) {
            System.out.println(e);
        }
    }
}
```

Output:

```
eProgram'
created table successfully...
```

115. Write a program to Update a record in the table.

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class DatabaseProgram {
    public static void main(String[] args) throws
Exception {
    try {
        //driver class

Class.forName("com.mysql.cj.jdbc.Driver");
        //creating a connection
        Connection con
=DriverManager.getConnection("jdbc:mysql://localhost:
3306/demo_class", "root", "root");
        System.out.println("Connected database
successfully.....");
        // creating a statement
        Statement stmt = con.createStatement();
        System.out.println("before update the
record");

        String sql1=("select *from student");
        ResultSet res=stmt.executeQuery(sql1);

        while (res.next()) {
            System.out.print("ID: " +
res.getInt("id"));
```

```

        System.out.print(",Name: " +
res.getString("name"));
        System.out.print(", Mobile Number: " +
res.getString("mobile"));
        System.out.println(", Institute: " +
res.getString("institue"));
    }
    //execute the query
    System.out.println("after update the
record");

    String query = ("UPDATE student SET name
= 'cccc' WHERE id = 4");
    stmt.executeUpdate(query);
    String sql=("select * from student");
    ResultSet rs=stmt.executeQuery(sql);

    while (rs.next()) {
        System.out.print("ID: " +
rs.getInt("id"));
        System.out.print(",Name: " +
rs.getString("name"));
        System.out.print(", Mobile Number: " +
rs.getString("mobile"));
        System.out.println(", Institute: " +
rs.getString("institue"));
    }
    //close the connection
    System.out.println("data updated
successfully");
    stmt.close();
} catch (Exception ex) {
    System.out.println(ex);
}

```

```
    }  
  }  
}
```

Output:

```
Connected database successfully.....  
before update the record  
ID: 1,Name: aaaa, Mobile Number: 1234567890, Institute: ocean academy  
ID: 2,Name: bbbb, Mobile Number: 1234569801, Institute:  
ID: 3,Name: ccc, Mobile Number: 1234567891, Institute: ocean academy  
ID: 4,Name: ddd, Mobile Number: 7448776015, Institute: ocean academy  
after update the record  
ID: 1,Name: aaaa, Mobile Number: 1234567890, Institute: ocean academy  
ID: 2,Name: bbbb, Mobile Number: 1234569801, Institute:  
ID: 3,Name: ccc, Mobile Number: 1234567891, Institute: ocean academy  
ID: 4,Name: cccc, Mobile Number: 7448776015, Institute: ocean academy  
data updated successfully
```

116. Write a program to Delete a table from the database.

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
public class DatabaseProgram {
    public static void main(String[] args) throws
Exception {
    try {
        //driver class
        Class.forName("com.mysql.cj.jdbc.Driver");
        //creating a connection
        Connection con
=DriverManager.getConnection("jdbc:mysql://localhost:
3306/demo_class", "root", "root");
        System.out.println("Connected database
successfully.....");
        // creating a statement
        Statement stmt = con.createStatement();
        String sql1=("Drop table student");
        stmt.executeUpdate(sql1);
        //close the connection
        System.out.println("table deleted
successfully");
        stmt.close();
    } catch (Exception ex) {
        System.out.println(ex);
    }
}
```



```
}
```

Output:

```
Connected database successfully.....  
table deleted successfully
```