

## Frequency Of character

1. Start
2. Define/read the string
3. Print "Enter the character"
4. Read the character
5. Set frequency = 0
6. loop( $i \leq \text{str.length}()$ )
7. if( $c == \text{str.charAt}(i)$ )
8. Increment frequency
9. Loop ends
10. Print frequency of character
11. Stop

## Matrix Multiplication

1. Start
2. Read no of rows (m) and column(n) for the first matrix ( $\text{mat1}[][]$ )
3. loop( $i < m$ )
  - a. loop( $j < n$ )  
Read matrix elements( $\text{mat1}[m][n]$ )
  - b. Loop ends
4. Loop ends
5. Read no of columns(p) for  $\text{mat2}[][]$  and no of rows will be equal to m
6. loop( $i < m$ )
  - c. loop( $j < p$ )  
Read matrix elements( $\text{mat1}[m][p]$ )
  - d. Loop ends
7. Loop ends
8. Declare new matrix  $\text{mul}[m][p]$  to store result
9. loop( $i < m$ )
  - loop( $j < p$ )
    - loop( $k < n$ )  
 $\text{Mul}[i][j] = \text{mul}[i][j] + \text{mat1}[i][k] * \text{mat2}[k][j]$
10. End all loops
11. Print resultant matrix
12. Stop

## Employee

1. Start
2. Create a class named 'Employee' with variables name, Age, phone, address, salary
3. Create method names printSalary to print salary details
4. Create a constructor named 'employee'
5. Create a method name displayEmployee() to display all details
6. Create a class named manager which extends employee class

7. Add variables specialization, department to manager class
8. Create a constructor named 'manager'
9. Create a method displayManager()
10. Inside displayManager() call displayEmployee and also include specialization and department
11. Create a new class name 'Officer' which extends employee class
12. Add variables- spclzation , department to officer class
13. Create a method displayOfficer() with displayEmployee() inside
14. In main() method
15. read manager details and display it
16. Read officer details and display it
17. Stop

## Abstract Class

1. Start
2. Create an abstract class named 'shape'
3. Inside that class create an abstract method named numberOfSides()
4. Create a class named "Rectangle" which extends Shape class
5. Inside the class, define numberOfSides() to print "Rectangle has four Sides"
6. Create a class named "Hexagon" which extends Shape class
7. Inside the class, define numberOfSides() to print "Hexagon has 6 Sides"
8. Create another class names "Triangle" which extends Shape class
9. Inside the class, define numberOfSides() to print "Triangle has 3 Sides"
10. Start main Method
11. Create objects of Rectangle, hexagon, and triangle
12. Call numberOfSides() for each individually
13. Stop

## File Handling

1. Start
2. Create objects of FileReader fr ,FileWriter fw and BuffferedReader br
3. Declare a string str
4. Loop ( (str == br.readLine()) != null)
5. Print string
6. fw.write(str) to copy.txt file
7. End loop
8. Close file reader
9. Cloise file writer
10. Handle exceptions if any
11. Stop

## String Tokenizer

1. Start
2. Import StringTokenizer package
3. Read the limit (n)
4. Read the numbers as a string names 'numList' separated by space
5. Create an object 'st' for StringTokenizer as new StringTokenizer(numList," ")
6. Set sum = 0
7. Loop while(st.hasMoreTokens())
8. Sum = sum + Integer.parseInt(st.nextToken())
9. End loop
10. Print sum
11. Stop

## Try, catch,throws

1. Start
2. Create a method divide(int a, intb)
3. If b == 0
4. Throw arithmetic exception and print("Division by 0 not possible")
5. else result = a/b
6. In main(), using try block read values for x, y and call divide()
7. Use catch block to handle arithmetic exception
8. And use finally block to print "End of program" even if division is not possible
9. Stop

## Thread Synchronization

1. Start
2. Create a synchronized method print(String msg)
3. print("[ " + msg)
4. In try block, call Thread.sleep() method
5. Use catch block to handle any InterruptedException
6. print("]")
7. Create class names SyncExample which extend Thread
8. Declare private Display d, private string msg
9. Create a method called run() and print msg using d.print(msg)
10. Create two objects of SyncExample with different msg
11. start() both threads
12. Stop

## Double linked List

1. Start
2. Declare three nodes head, left, right
3. Create a method named "InsertAtEnd"
4. And create a temp node
5. if(head == null)
6. Head == temp
7. If ends
8. Else
9. Node ptr = head
10. Loop(ptr.right != null)
11. Ptr = ptr.right
12. Else ends
13. Ptr.right = temp;
14. Temp.right = ptr
15. Loop ends
16. Print data inserted successfully
17. Create a method named 'deleteAtFront'
18. if(head == null)
19. Print empty list
20. Else
21. Int data = head.data
22. Head = head.right
23. Head.left = null
24. Print data is deleted
25. Create another method display()
26. Node temp = head
27. if(head == null)
28. Print empty list
29. Else
30. loop(temp != null)
31. Print temp.data
32. Temp = temp.right
33. Loop ends
34. Create a menu to select b/w insert delete or display
35. Stop

## Quick sort

1. Start
2. Create a method quickSort(string A[], int p, int r)
3. if(p < r)
4. Q = partition(A, p, r)
5. quickSort(A, p, q-1)
6. quickSort(A, q+1, r)

7. If ends
8. Create another method partition(String[], int p, int r)
9. String x = A[r]
10. Int i = p-1
11. Loop (j=p;j<=r-1)
12. if(a[j].compareTo(x) <= 0)
13. I = i+1
14. String temp = A[j]
15. A[i] = a[j]
16. a [j] = temp
17. If ends
18. Loops ends
19. String temp = A[r +1]
20. A[i+1] = A[r]
21. A[r] = temp
22. Return i+1
23. Read the limit (n)
24. loop(i<n)
25. Read the string ( A[i] = sc.next:ine())
26. Loop ends
27. quickSort(A,0,n-1)
28. Print sorted array
29. Stop

## Basic Programs

### Triangle

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

### Algorithm

1. Start
2. Set i=0, j=0;
3. Read no of rows(row)
4. Loop (i<row)
5. loop(j<=i)
6. Print "\*"
7. Loop ends
8. println("")
9. Loop ends
10. Stop

## Program

```
public class RightTrianglePattern {
    public static void main(String args[]) {
        int i, j, row=6;
        for(i=0; i<row; i++) {
            for(j=0; j<=i; j++)
                {
                    System.out.print("* ");
                }
            System.out.println();
        }
    }
}
```

## Pyramid triangle

```

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

## Algorithm

1. Start
2. Set i=0,j=0
3. Read no of rows(row)
4. loop(i<row)
5. loop(j=row-1;j>1;j--)
6. Print " "
7. Loop ends
8. loop(j<=i)
9. Print "\*"
10. Loop ends
11. Println()
12. Loop ends
13. Stop

## Program

```
public class PyramidPattern {
    public static void main(String args[]) {
        int i, j, row = 6;
        for (i=0; i<row; i++) {
            for (j=row-i; j>1; j--) {
                System.out.print(" ");
            }
        }
    }
}
```

```

    for (j=0; j<=i; j++ ) {
        System.out.print("* ");
    }
    System.out.println();
}
}
}

```

## **Fibonacci**

0 1 1 2 3 5 8 13 21 34

## **Algorithm**

1. Start
2. Set n1=0,n2 =1
3. Set limit to 10(or as required)
4. Print 0 1
5. loop(i=2;i<limit)
6. n3 = n2+n1
7. Print " " + n3
8. n1 = n2
9. n2 = n3
10. Loop ends
11. Stop

## **Program**

```

class Fibonacci{
    public static void main(String args[]) {
        int n1=0,n2=1,n3,i,count=10;
        System.out.print(n1+" "+n2);
        for(i=2;i<count;++i) {
            n3=n1+n2;
            System.out.print(" "+n3);
            n1=n2;
            n2=n3;
        }
    }
}

```

## **Palindrome Number**

## **Algorithm**

1. Start
2. Set sum =0
3. Read number

4. Set temp = n
5. loop(n>0)
6. r = n%10
7. sum = (sum\*10)+r
8. n = n/10
9. Loop ends
- 10.If (temp == sum)
- 11.Print palindrome
- 12.Else
- 13.Print not palindrome
14. Stop

### **Program**

```
class PalindromeExample{
public static void main(String args[]){
    int r,sum=0,temp;
    int n=454;
    temp=n;
    while(n>0){
        r=n%10; //getting remainder
        sum=(sum*10)+r;
        n=n/10;
    }
    if(temp==sum)
        System.out.println("palindrome number ");
    else
        System.out.println("not palindrome");
    }
}
```

### **Palindrome string**

#### **Algorithm**

1. Start
2. Set str , rev = " "
3. Print enter a string
4. Read the string
5. Length = str.length()
6. loop(i= length -1; i>=0; i--)
7. rev = rev + str.charAt(i);
8. Loop ends
9. if (str.equals(rev))
10. Print palindrome
11. Else
12. Print not palindrome
13. Stop



## Program

```
import java.util.Scanner;
class ChkPalindrome{
    public static void main(String args[]) {
        String str, rev = "";
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string:");
        str = sc.nextLine();
        int length = str.length();
        for ( int i = length - 1; i >= 0; i-- )
            rev = rev + str.charAt(i);
        if (str.equals(rev))
            System.out.println(str+" is a palindrome");
        else
            System.out.println(str+" is not a palindrome");
    }
}
```

**!!!!!!! completed Calculator, traffic light not included !!!!!!!**