1. public class Main {

```
static int sumRec(int n) {
    if (n == 0) return 0;
    return (n % 10) + sumRec(n / 10);
}
public static void main(String[] args) {
    int num = 1234;
    int temp = num, sum = 0;
    while (temp != 0) {
        sum += temp % 10;
        temp /= 10;
    }
    System.out.println("Sum using loop = " + sum);
    System.out.println("Sum using recursion = " + sumRec(num));
}
}
```

2. public class Main {

```
static int fib(int n) {
    if (n <= 1) return n;
    return fib(n-1) + fib(n-2);
}
public static void main(String[] args) {
    int N = 7;
    for (int i = 0; i < N; i++) {
        System.out.print(fib(i) + " ");       }}}
```

3. public class Main {

public static void main(String[] args) {

int n = 5;

boolean prime = true;

if (n <= 1)

prime = false;

else {

for (int i = 2; i <= n / 2; i++) {

if (n % i == 0) {

prime = false;

break;

} } }

System.out.println(n + (prime ? " is Prime" : " is Not Prime")); } }

4. public class Main {

public static void main(String[] args) {

int n = 4;

for (int i = n; i >= 1; i--) {

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

System.out.print(" ");

for (int j = i; j >= 1; j--)

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

System.out.println();

} } }

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

6.
```
public class Main {
public static void main(String[] args) {
    for (int i = 1; i <= 5; i++) {
        for (int j = 1; j <= 5; j++) {
            if ((i == j) || (i + j == 6)) {
                System.out.print("* ");
            } else {
                System.out.print("  ");
            }
        }
        System.out.println();
    }
}
}
```

7.
```
public class Main {
  public static void main(String[] args) {
    int arr[] = {1, 2, 3, 4, 5};
    int start = 0;
    int end = arr.length - 1;
    while (start < end) {
      int t = arr[start];
      arr[start] = arr[end];
      arr[end] = t;
      start++;
      end--;
    }
    for (int x : arr)
      System.out.print(x + " ");
  }
}
```

8.
```
public class Main {
  public static void main(String[] args) {
    int arr[] = {10, 40, 20, 50, 30};
    int max = Integer.MIN_VALUE;
    int second = Integer.MIN_VALUE;
    for (int n : arr) {
      if (n > max) {
        second = max;
        max = n;
      } else if (n > second && n != max) {
```

```
            second = n;
           } }
System.out.println("Second largest = " + second);
 }}
```

9.
```
import java.util.*;
 public class Main {
public static void main(String[] args) {
 int a[] = {1, 3, 5};
 int b[] = {2, 4, 6};
 int c[] = new int[a.length + b.length];
 int k = 0;
 for (int i = 0; i < a.length; i++)
    c[k++] = a[i];
 for (int j = 0; j < b.length; j++)
    c[k++] = b[j];
 Arrays.sort(c);
 for (int x : c)
    System.out.print(x + " ");
  }
}
```

```
10.class BankAccount {
 private double balance;
 void deposit(double amt) {
    balance += amt;
  }
  void withdraw(double amt) {
    if (amt <= balance)
    balance -= amt;
  }
  double getBalance() {
 return balance;
  }
}
public class Main {
  public static void main(String[] args) {
    BankAccount b = new BankAccount();
    b.deposit(1000);
    b.withdraw(300);
    System.out.println("Balance = " + b.getBalance());
  }
}
```

```
11.class Employee {
   int id;
   String name;
   String dept;
   double basic;
   Employee(int id, String name, String dept, double basic) {
      this.id = id;
      this.name = name;
      this.dept = dept;
      this.basic = basic;
   }
   double calculateSalary() {
      return basic;
   }
}
class Developer extends Employee {
   double allowance;
   Developer(int id, String name, String dept, double basic, double allowance)
{
      super(id, name, dept, basic);
      this.allowance = allowance;
   }
   double calculateSalary() {
      return basic + allowance;
   }
}
```

```
class Tester extends Employee {
    double bonus;
    Tester(int id, String name, String dept, double basic, double bonus) {
        super(id, name, dept, basic);
        this.bonus = bonus;
    }

    double calculateSalary() {
        return basic + bonus;
    }
}

     public class Main {
    public static void main(String[] args) {
        Employee e1 = new Developer(1, "Raghu", "Dev", 40000, 8000);
        Employee e2 = new Tester(2, "Naik", "QA", 35000, 5000);
        System.out.println(e1.name + " Salary: " + e1.calculateSalary());
        System.out.println(e2.name + " Salary: " + e2.calculateSalary());}}

     12.public class Main {
    public static void main(String[] args) {
        String s = "Hello@123";
        int v=0, c=0, d=0, sp=0;

        for (char ch : s.toCharArray()) {
            if (Character.isDigit(ch))
```

```
            d++;
        else if ("AEIOUaeiou".indexOf(ch) != -1)
            v++;
        else if (Character.isLetter(ch))
            c++;
        else
            sp++;
      }
      System.out.println("Vowels: " + v);
      System.out.println("Consonants: " + c);
      System.out.println("Digits: " + d);
      System.out.println("Special: " + sp);
   }
}
```

13.

```
import java.util.Arrays;
public class Main {
   public static void main(String[] args) {
      String a = "silent";
      String b = "listen";
      char x[] = a.toCharArray();
      char y[] = b.toCharArray();
      Arrays.sort(x);
      Arrays.sort(y);
      System.out.println(Arrays.equals(x, y) ? "Anagram" : "Not Anagram");
   }
}
```

14.public class Main {
public static void main(String[] args) {
String s = "Java is Easy";
String words[] = s.split(" ");

for (String w : words) {
String rev = "";
for (int i = w.length()-1; i >= 0; i--)
rev += w.charAt(i);
System.out.print(rev + " "); } } }

15.import java.util.*;
public class Main {
public static void main(String[] args) {
List<Integer> list = Arrays.asList(1,2,2,3,3,4);
Set<Integer> set = new HashSet<>(list);
System.out.println(set);
}
}

16.import java.util.*;
public class Main {
public static void main(String[] args) {
String s = "hello";
HashMap<Character, Integer> map = new HashMap<>();
for (char ch : s.toCharArray())
map.put(ch, map.getOrDefault(ch, 0) + 1);

```
        System.out.println(map);
    }
}
```

17.

```
import java.util.*;
 public class Main {
    public static void main(String[] args) {
    TreeSet<Integer> set = new TreeSet<>();
    set.add(40);
    set.add(10);
    set.add(20);
    set.add(30);
    System.out.println("Ascending: " + set);
    System.out.println("Descending: " + set.descendingSet());
    }
}
```

18.

```
class EvenThread extends Thread {
    public void run() {
        for (int i = 2; i <= 50; i += 2) {
            System.out.print(i + " ");
        }
    }
}
class OddThread extends Thread {
    public void run() {
        for (int i = 1; i <= 49; i += 2) {
            System.out.print(i + " ");
```

```
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Thread even = new EvenThread();
        Thread odd = new OddThread();
        even.start();
        odd.start();
    }
}
     19.import java.util.*;
import java.util.stream.*;
public class Main {
    public static void main(String[] args) {
        List<Integer> list = Arrays.asList(1,2,3,4,5,6);
        List<Integer> evens = list.stream()
                            .filter(n -> n % 2 == 0)
                            .collect(Collectors.toList());
        System.out.println(evens);
    }
}
class InvalidAgeException extends Exception {
    InvalidAgeException(String msg) {
        super(msg); } }
```

```
20.public class Main {

static void checkAge(int age) throws InvalidAgeException {
    if (age < 18) {
        throw new InvalidAgeException("Not eligible to vote");
    }
    System.out.println("Eligible to vote");
}

public static void main(String[] args) {

    try {
        checkAge(16);
    } catch (InvalidAgeException e) {
        System.out.println(e.getMessage());
    }
}
}
```