```
================================================================================
Complex number and Imaginary part….
================================================================================
import java.util.Scanner;

class Complex_No {
   float real, imag;

   // Constructor to initialize the complex number
   public Complex_No(float real, float imag) {
      this.real = real;
      this.imag = imag;
   }

   // Method to display the complex number
   public void D() {
      System.out.println("(" + real + ") + (" + imag + ")i");
   }

   // Addition of two complex numbers
   public Complex_No A(Complex_No other) {
      return new Complex_No(this.real + other.real, this.imag + other.imag);
   }

   // Subtraction of two complex numbers
   public Complex_No B(Complex_No other) {
      return new Complex_No(this.real - other.real, this.imag - other.imag);
   }

   // Multiplication of two complex numbers
   public Complex_No C(Complex_No other) {
      float realPart = this.real * other.real - this.imag * other.imag;
      float imagPart = this.real * other.imag + this.imag * other.real;
      return new Complex_No(realPart, imagPart);
   }

   // Division of two complex numbers
   public Complex_No Dv(Complex_No other) {
      float denominator = other.real * other.real + other.imag * other.imag;
      if (denominator == 0) {
         System.out.println("Error: Division by zero.");
         return null;
      }
      float realPart = (this.real * other.real + this.imag * other.imag) / denominator;
```

```
        float imagPart = (this.imag * other.real - this.real * other.imag) / denominator;
        return new Complex_No(realPart, imagPart);
    }
}

public class ComplexNumberOperations {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        // Input for first complex number
        System.out.print("Enter real part of 1st complex number: ");
        float real1 = scan.nextFloat();
        System.out.print("Enter imaginary part of 1st complex number: ");
        float imag1 = scan.nextFloat();
        Complex_No C1 = new Complex_No(real1, imag1);

        // Input for second complex number
        System.out.print("Enter real part of 2nd complex number: ");
        float real2 = scan.nextFloat();
        System.out.print("Enter imaginary part of 2nd complex number: ");
        float imag2 = scan.nextFloat();
        Complex_No C2 = new Complex_No(real2, imag2);

        scan.close();

        // Display the complex numbers
        System.out.print("1st Complex Number: ");
        C1.D();
        System.out.print("2nd Complex Number: ");
        C2.D();

        // Perform and display operations
        Complex_No result;

        result = C1.A(C2);  // Addition
        System.out.print("Addition: ");
        result.D();

        result = C1.B(C2);  // Subtraction
        System.out.print("Subtraction: ");
        result.D();

        result = C1.C(C2);  // Multiplication
        System.out.print("Multiplication: ");
```

```
        result.D();

        result = C1.Dv(C2); // Division
        if (result != null) {
            System.out.print("Division: ");
            result.D();
        }
    }
}
```

==========

Identify commonalities and differences between Publication, Book and Magazine classes.
Title, Price, Copies are common instance variables and saleCopy is common method.
The differences are, Bookclass has author and orderCopies().
Magazine Class has data members as orderQty, Current issue and method receiveNewissue().Write a program to find how many copies of the given books are ordered and display total sale of publication.

==========

```
import java.util.Scanner;
class Publication {
    String title;
    int price;
    int copies;

    // Method to read publication details
    public void readDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter title: ");
        title = sc.nextLine();
        System.out.print("Enter price: ");
        price = sc.nextInt();
        System.out.print("Enter number of copies: ");
        copies = sc.nextInt();
    }

    // Method to display publication details
    public void displayDetails() {
        System.out.println("Title: " + title + ", Price: " + price + ", Copies available: " + copies);
    }

    // Method to sell copies of the publication
    public void sellCopies(int number) {
        if (number <= copies) {
            copies -= number;
            System.out.println("You bought " + number + " copies. Total cost: " + (number * price));
```

```
        } else {
            System.out.println("Not enough copies in stock.");
        }
    }

    // Method to order more copies
    public void orderCopies(int number) {
        copies += number;
        System.out.println("Successfully ordered " + number + " new copies.");
    }
}

public class Main extends Publication
{

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

        Publication book = new Publication();
        Publication magazine = new Publication();

        int choice;
        do {
            System.out.println("\nChoose one of the following options:");
            System.out.println("1. Book");
            System.out.println("2. Magazine");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1: // Manage Book
                    System.out.println("\nManaging Book:");
                    managePublication(book);
                    break;

                case 2: // Manage Magazine
                    System.out.println("\nManaging Magazine:");
                    managePublication(magazine);
                    break;

                case 3: // Exit
                    System.out.println("Exiting program...");
                    break;
```

```
            default:
                System.out.println("Invalid choice, please try again.");
                break;
        }

    } while (choice != 3);

    sc.close();  // Close the scanner to prevent memory leaks
}

// Method to handle book and magazine actions
public static void managePublication(Publication publication) {
    Scanner sc = new Scanner(System.in);

    // Read publication details
    publication.readDetails();

    int action;
    do {
        System.out.println("\nChoose an action:");
        System.out.println("1. Display details");
        System.out.println("2. Sell copies");
        System.out.println("3. Order more copies");
        System.out.println("4. Go back to main menu");
        System.out.print("Enter your choice: ");
        action = sc.nextInt();

        switch (action) {
            case 1:
                publication.displayDetails();
                break;

            case 2:
                System.out.print("Enter number of copies to sell: ");
                int sellCount = sc.nextInt();
                publication.sellCopies(sellCount);
                break;

            case 3:
                System.out.print("Enter number of copies to order: ");
                int orderCount = sc.nextInt();
                publication.orderCopies(orderCount);
                break;
```

```
            case 4:
                System.out.println("Going back to main menu...");
                break;

            default:
                System.out.println("Invalid choice, please try again.");
                break;
        }

    } while (action != 4);
  }
}
```

==========
OUTPUT:
==========

```
Choose one of the following options:
1. Book
2. Magazine
3. Exit
Enter your choice: 1

Managing Book:
Enter title: aaa
Enter price: 100
Enter number of copies: 2

Choose an action:
1. Display details
2. Sell copies
3. Order more copies
4. Go back to main menu
Enter your choice: 1
Title: aaa, Price: 100, Copies available: 2

Choose an action:
1. Display details
2. Sell copies
3. Order more copies
4. Go back to main menu
Enter your choice: 2
Enter number of copies to sell: 3
Not enough copies in stock.
```

Choose an action:
1. Display details
2. Sell copies
3. Order more copies
4. Go back to main menu
Enter your choice: 3
Enter number of copies to order: 1
Successfully ordered 1 new copies.

Choose an action:
1. Display details
2. Sell copies
3. Order more copies
4. Go back to main menu
Enter your choice: 4
Going back to main menu...

Choose one of the following options:
1. Book
2. Magazine
3. Exit
Enter your choice: 2

Managing Magazine:
Enter title: dddd
Enter price: 200
Enter number of copies: 1

Choose an action:
1. Display details
2. Sell copies
3. Order more copies
4. Go back to main menu
Enter your choice: 1
Title: dddd, Price: 200, Copies available: 1

Choose an action:
1. Display details
2. Sell copies
3. Order more copies
4. Go back to main menu
Enter your choice: 2
Enter number of copies to sell: 3

Not enough copies in stock.

Choose an action:
1. Display details
2. Sell copies
3. Order more copies
4. Go back to main menu
Enter your choice: 1
Title: dddd, Price: 200, Copies available: 1

Choose an action:
1. Display details
2. Sell copies
3. Order more copies
4. Go back to main menu
Enter your choice: 4
Going back to main menu...

Choose one of the following options:
1. Book
2. Magazine
3. Exit
Enter your choice: 3
Exiting program...

...Program finished with exit code 0
Press ENTER to exit console.

==========

Design and develop inheritance for a given case study, identify objects and relationships and implement inheritance wherever applicable. Employee class has Emp_name, Emp_id, Address, Mail_id, and Mobile_no as members. Inherit the classes: Programmer, Team Lead, Assistant Project Manager and Project Manager from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.

==========

```
import java.util.Scanner;

class Employee {
   int empid;
   long mobile;
```

```
String name, address, mailid;
double basic, da, hra, pf, gross, net;

// Constructor to initialize salary
Employee(double salary) {
   basic = salary;
}

// Method to input employee details
void getdata() {
   Scanner sc = new Scanner(System.in);

   System.out.println("Enter Name of the Employee: ");
   name = sc.nextLine();

   System.out.println("Enter Mail ID: ");
   mailid = sc.nextLine();

   System.out.println("Enter Address of the Employee: ");
   address = sc.nextLine();

   System.out.println("Enter Employee ID: ");
   empid = sc.nextInt();

   System.out.println("Enter Mobile Number: ");
   mobile = sc.nextLong();
}

// Method to calculate salary details
void calculateSalary() {
   da = 0.97 * basic;  // Dearness Allowance (97% of basic)
   hra = 0.10 * basic; // House Rent Allowance (10% of basic)
   pf = 0.12 * basic;  // Provident Fund (12% of basic)

   gross = basic + da + hra;   // Gross salary = basic + DA + HRA
   net = gross - pf;           // Net salary = gross - PF

   // Display salary details
   System.out.println("\nGROSS PAY: Rs. " + gross);
   System.out.println("NET PAY: Rs. " + net);
}

// Method to display employee details
void display() {
```

```
        System.out.println("\nEmployee Name: " + name);
        System.out.println("Employee ID: " + empid);
        System.out.println("Mail ID: " + mailid);
        System.out.println("Address: " + address);
        System.out.println("Mobile Number: " + mobile);
    }
}

public class SimplePayroll {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice, cont=0;

        do {
            // Display role options
            System.out.println("\nPAYROLL SYSTEM");
            System.out.println("1. Programmer");
            System.out.println("2. Team Leader");
            System.out.println("3. Assistant Project Manager");
            System.out.println("4. Project Manager");
            System.out.print("Enter Your Choice (1-4): ");
            choice = sc.nextInt();

            // Handle the different choices
            switch (choice) {
                case 1:
                    Employee p = new Employee(30000); // Programmer with 30,000 salary
                    p.getdata();
                    p.display();
                    p.calculateSalary();
                    break;
                case 2:
                    Employee t = new Employee(40000); // Team Leader with 40,000 salary
                    t.getdata();
                    t.display();
                    t.calculateSalary();
                    break;
                case 3:
                    Employee a = new Employee(50000); // Assistant Project Manager with 50,000 salary
                    a.getdata();
                    a.display();
                    a.calculateSalary();
                    break;
                case 4:
```

```
                Employee m = new Employee(70000); // Project Manager with 70,000 salary
                m.getdata();
                m.display();
                m.calculateSalary();
                break;
            default:
                System.out.println("Invalid choice! Please enter a number between 1 and 4.");
        }

        // Ask if user wants to continue
        System.out.print("Do you want to continue? (1 for Yes, 0 for No): ");
        cont = sc.nextInt();
    } while (cont == 1);  // Repeat if user chooses 1 to continue

    System.out.println("Exiting the Payroll System. Goodbye!");
  }
}
```

==========

OUTPUT:

==========

```
PAYROLL SYSTEM
1. Programmer
2. Team Leader
3. Assistant Project Manager
4. Project Manager
Enter Your Choice (1-4): 1
Enter Name of the Employee: aaa
Enter Mail ID: aa@123
Enter Address of the Employee: jalna
Enter Employee ID: 1010
Enter Mobile Number: 8767419772

Employee Name: aaa
Employee ID: 1010
Mail ID: aa@123
Address: jalna
Mobile Number: 8767419772

GROSS PAY: Rs. 62100.0
NET PAY: Rs. 58500.0
Do you want to continue? (1 for Yes, 0 for No): 1

PAYROLL SYSTEM
1. Programmer
```

2. Team Leader
3. Assistant Project Manager
4. Project Manager
Enter Your Choice (1-4): 2
Enter Name of the Employee: bbb
Enter Mail ID: bb@123
Enter Address of the Employee: jalna
Enter Employee ID: 2020
Enter Mobile Number: 9783476573

Employee Name: bbb
Employee ID: 2020
Mail ID: bb@123
Address: jalna
Mobile Number: 9783476573

GROSS PAY: Rs. 82800.0
NET PAY: Rs. 78000.0
Do you want to continue? (1 for Yes, 0 for No): 1

PAYROLL SYSTEM
1. Programmer
2. Team Leader
3. Assistant Project Manager
4. Project Manager
Enter Your Choice (1-4): 3
Enter Name of the Employee: ccc
Enter Mail ID: cc@123
Enter Address of the Employee: pune
Enter Employee ID: 3030
Enter Mobile Number: 8945789533

Employee Name: ccc
Employee ID: 3030
Mail ID: cc@123
Address: pune
Mobile Number: 8945789533

GROSS PAY: Rs. 103500.0
NET PAY: Rs. 97500.0
Do you want to continue? (1 for Yes, 0 for No): 1

PAYROLL SYSTEM
1. Programmer

2. Team Leader
3. Assistant Project Manager
4. Project Manager
Enter Your Choice (1-4): 4
Enter Name of the Employee: dddd
Enter Mail ID: d@123
Enter Address of the Employee: pune
Enter Employee ID: 4040
Enter Mobile Number: 9476783452

Employee Name: dddd
Employee ID: 4040
Mail ID: d@123
Address: pune
Mobile Number: 9476783452

GROSS PAY: Rs. 144900.0
NET PAY: Rs. 136500.0
Do you want to continue? (1 for Yes, 0 for No): 0
Exiting the Payroll System. Goodbye!

...Program finished with exit code 0
Press ENTER to exit console.

==========

Design a base class shape with two double type values and member function to input the data and compute area() for calculating area of shape. Derive two classes Triangle and Rectangle make compute area() as abstract function and redefine this function in the derive class to suit their requirement .Write a program that accepts the dimension of Triangle /Rectangle and display calculated area. Implement dynamic binding.

***/**

==========

```
import java.util.Scanner;

class Shape {
   double val1, val2;

   // Method to get input
   void input() {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter first value: ");
```

```
        val1 = s.nextDouble();
        System.out.print("Enter second value: ");
        val2 = s.nextDouble();
    }
}

class Triangle extends Shape {
    // Method to calculate area of the triangle
    void computeArea() {
        double area = 0.5 * val1 * val2; // Area = 1/2 * base * height
        System.out.println("Triangle Area: " + area);
    }
}

class Rectangle extends Shape {
    // Method to calculate area of the rectangle
    void computeArea() {
        double area = val1 * val2; // Area = length * width
        System.out.println("Rectangle Area: " + area);
    }
}

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

        // For Triangle
        Triangle triangle = new Triangle();
        System.out.println("Enter dimensions for Triangle:");
        triangle.input();
        triangle.computeArea();

        // For Rectangle
        Rectangle rectangle = new Rectangle();
        System.out.println("Enter dimensions for Rectangle:");
        rectangle.input();
        rectangle.computeArea();
    }
}
```

==========
OUTPUT:
==========

```
Enter dimensions for Triangle:
Enter first value: 10
Enter second value: 20
Triangle Area: 100.0
Enter dimensions for Rectangle:
Enter first value: 20
Enter second value: 12
Rectangle Area: 240.0


...Program finished with exit code 0
Press ENTER to exit console.
```

==========

*** Exception handling**

Implement a program to handle Arithmetic exception, Array Index Out of Bounds. The user enters two numbers Num1 and Num2. The division of Num1 and Num2 is displayed. If Num1 and Num2 are not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception. Display the exception. */

==========

```
import java.util.Scanner;

class ExceptionHandlingDemo {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int choice;

      do {
         System.out.println("\nChoose an exception to handle:");
         System.out.println("1. ArithmeticException (Divide by Zero)");
         System.out.println("2. ArrayIndexOutOfBoundsException");
         System.out.println("3. NumberFormatException (Invalid Integer)");
         System.out.println("4. Exit");
         System.out.print("Enter your choice: ");
         choice = sc.nextInt();

         switch (choice) {
            case 1:
               try {
                  System.out.print("Enter first number: ");
                  int num1 = sc.nextInt();
```

```
                System.out.print("Enter second number: ");
                int num2 = sc.nextInt();
                int result = num1 / num2; // Division that may cause ArithmeticException
                System.out.println("Result: " + result);
            } catch (ArithmeticException e) {
                System.out.println("Error: Cannot divide by zero!");
            }
            break;

        case 2:
            try {
                int[] arr = new int[5];
                arr[2] = 10;
                System.out.println("Value at index 2: " + arr[2]);
                arr[10] = 20; // This will cause ArrayIndexOutOfBoundsException
            } catch (ArrayIndexOutOfBoundsException e) {
                System.out.println("Error: Array index out of bounds!");
            }
            break;

        case 3:
            try {
                String str = "XYZ";
                int number = Integer.parseInt(str); // This will cause NumberFormatException
                System.out.println("Parsed number: " + number);
            } catch (NumberFormatException e) {
                System.out.println("Error: Invalid number format!");
            }
            break;

        case 4:
            System.out.println("Exiting the program.");
            break;

        default:
            System.out.println("Invalid choice. Please try again.");
        }

    } while (choice != 4); // Loop until the user chooses to exit
  }
}
```

```
========================
OUTPUT:
========================
Choose an exception to handle:
1. ArithmeticException (Divide by Zero)
2. ArrayIndexOutOfBoundsException
3. NumberFormatException (Invalid Integer)
4. Exit
Enter your choice: 1
Enter first number: 10
Enter second number: 0
Error: Cannot divide by zero!

Choose an exception to handle:
1. ArithmeticException (Divide by Zero)
2. ArrayIndexOutOfBoundsException
3. NumberFormatException (Invalid Integer)
4. Exit
Enter your choice: 2
Value at index 2: 10
Error: Array index out of bounds!

Choose an exception to handle:
1. ArithmeticException (Divide by Zero)
2. ArrayIndexOutOfBoundsException
3. NumberFormatException (Invalid Integer)
4. Exit
Enter your choice: 3
Error: Invalid number format!

Choose an exception to handle:
1. ArithmeticException (Divide by Zero)
2. ArrayIndexOutOfBoundsException
3. NumberFormatException (Invalid Integer)
4. Exit
Enter your choice: 4
Exiting the program.


...Program finished with exit code 0
Press ENTER to exit console.
```

==========

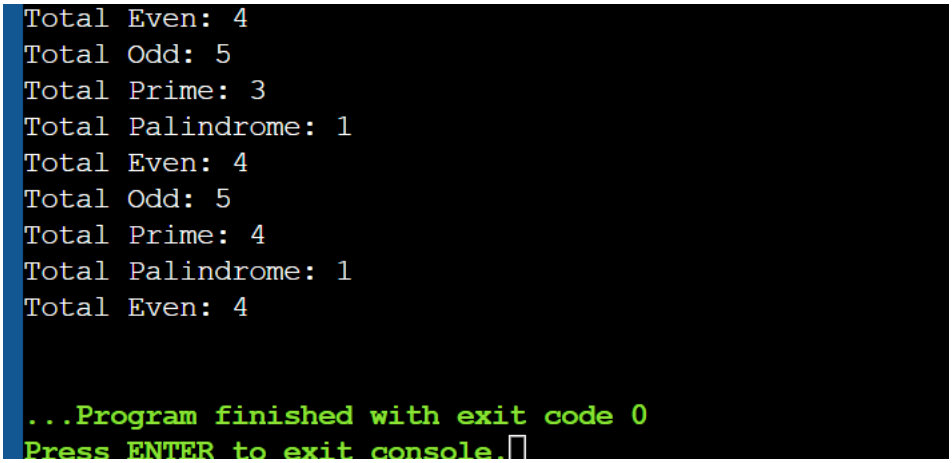
Implement a generic program using any collection class to count the number of elements in a collection that have a specific property such as even numbers, odd number, prime number and palindromes.*/

==========

```
public class Main {

   // Method to check if a number is prime
   static boolean isPrime(int num) {
      if (num <= 1) return false;
      for (int i = 2; i <= Math.sqrt(num); i++) {
         if (num % i == 0) return false;
      }
      return true;
   }

   // Generic method to count even, odd, prime, or palindrome numbers
   static <T> void count(String type, T[] elements) {
      int even = 0, odd = 0, prime = 0, palin = 0;

      // Loop through the elements and perform checks based on the type
      for (T element : elements) {
         String valueStr = element.toString();
         int value = Integer.parseInt(valueStr);

         switch (type) {
            case "even":
               if (value % 2 == 0) even++;
               break;
            case "odd":
               if (value % 2 != 0) odd++;
               break;
            case "prime":
               if (isPrime(value)) prime++;
               break;
            case "palindrome":
               if (valueStr.equals(new StringBuilder(valueStr).reverse().toString())) palin++;
               break;
         }
      }

      // Print the result based on the type
      switch (type) {
         case "even":
```

```
            System.out.println("Total Even: " + even);
            break;
        case "odd":
            System.out.println("Total Odd: " + odd);
            break;
        case "prime":
            System.out.println("Total Prime: " + prime);
            break;
        case "palindrome":
            System.out.println("Total Palindrome: " + palin);
            break;
      }
   }

   public static void main(String[] args) {
      // Test arrays of different numeric types
      Integer[] iarray = {45, 7, 12, 84, 38, 115, 29, 30, 19};
      count("even", iarray);
      count("odd", iarray);
      count("prime", iarray);
      count("palindrome", iarray);

      // Test arrays with other types (Byte, Short, Long)
      Byte[] barray = {45, 7, 12, 84, 38, 115, 29, 30, 19};
      count("even", barray);

      Short[] sarray = {45, 73, 12, 84, 38, 151, 29, 30, 19};
      count("odd", sarray);
      count("prime", sarray);
      count("palindrome", sarray);

      Long[] larray = {45L, 73L, 12L, 84L, 38L, 151L, 29L, 30L, 19L};
      count("even", larray);
   }
}
```

==========
OUTPUT:
==========


Total Even: 4
Total Odd: 5
Total Prime: 3
Total Palindrome: 1
Total Even: 4
Total Odd: 5
Total Prime: 4
Total Palindrome: 1
Total Even: 4
...Program finished with exit code 0
Press ENTER to exit console.