-Additional Java Programs-

1. Write a Java Program to create an abstract class named sum of Two and sum of Three. Perform addition of two numbers and addition of three numbers

Coding :

```
abstract class Sum {
    abstract void sumOfTwo(int n1, int n2);
    abstract void sumOfThree(int n1, int n2, int n3);
}

class Addition extends Sum {
    void sumOfTwo(int n1, int n2) {
        System.out.println("Sum of " + n1 + " and " + n2 + " is: " + (n1 +
n2));
    }
    void sumOfThree(int n1, int n2, int n3) {
        System.out.println("Sum of " + n1 + ", " + n2 + " and " + n3 + " is: "
+ (n1 + n2 + n3));
    }
}

class Main {
    public static void main(String[] args) {
        Addition add = new Addition();
        add.sumOfTwo(5, 7);
        add.sumOfThree(5, 7, 3);
    }
}
```

2.Write a program to Check Prime Number using Interface.

Coding :

```
import java.util.Scanner;
```

```
interface PrimeCheck {
    boolean isPrime(int num);
}

class PrimeNumberCheck implements PrimeCheck {
    public 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;
    }
}

class Main {
    public static void main(String[] args) {
        PrimeNumberCheck primeCheck = new PrimeNumberCheck();
        Scanner in = new Scanner(System.in);
        int num;
        System.out.println("enter a number to check : ");
        num = in.nextInt();
        if (primeCheck.isPrime(num)) {
            System.out.println(num + " is a prime number.");
        } else {
            System.out.println(num + " is not a prime number.");
        }
    }
}
```

3. Write a Java program to find the sum value of two given type of elements using a generic class.

Coding :

```
class Addition<T> {
    T num1;
    T num2;
    Addition(T num1, T num2) {
        this.num1 = num1;
        this.num2 = num2;
    }
    public void add() {
        if(num1 instanceof Integer && num2 instanceof Integer) {
            System.out.println("The sum of the integers is: " + ((Integer)num1 + (Integer)num2));
        }
        else if(num1 instanceof Double && num2 instanceof Double) {
            System.out.println("The sum of the doubles is: " + ((Double)num1 + (Double)num2));
        }
        else {
            System.out.println("Invalid input");
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Addition<Integer> iob = new Addition<Integer>(5, 10);
        iob.add();
        Addition<Double> dob = new Addition<Double>(5.5, 10.5);
        dob.add();
    }
}
```

5. Write Java programs to implementing Arithmetic exception and implementing Array IndexOutOfBound exception.

Coding :

>> Arithmetic Exception <<

```
class Arithmetic {
    public static void main(String[] args) {
        try {
            int num1 = 30, num2 = 0;
            int result = num1 / num2;
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Cannot divide by zero: " + e);
        }
    }
}
```

>>Array Out of Bound Exception <<

```
class ArrayIndexOutOfBounds {
    public static void main(String[] args) {
        try {
            int[] arr = new int[5];
            arr[6] = 10;
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Array index out of bound: " + e);
        }
    }
}
```