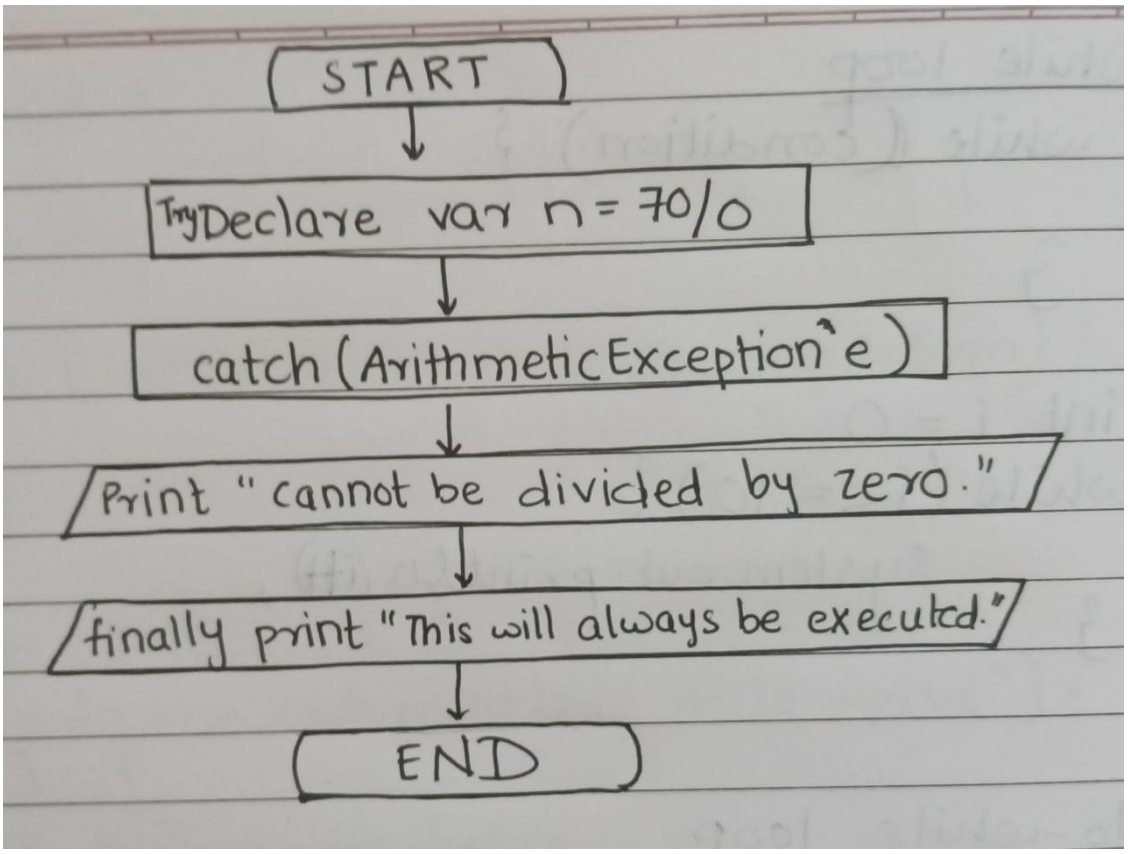
ASSIGNMENT 7

Java Program 1: Write program using try, catch and finally. using to give the output “Cannot be divided by zero” and “This will be always executed.”

Algorithm:

1. Define a variable n.
2. Now in the try block write the code, that is declare n=70/0 and print the number.
3. In catch(ArithmeticException e), print the Exception.
4. Then in finally, print the statement “This always needs to be executed.”

Flowchart:



Code:

```
package exception;

public class Program1 {
	public static void main(String[] args) {
		int n;

		try{
			n = 70/0;
		}
		catch(ArithmeticException e){
			System.out.println("Cannot be divided by Zero.");
		}
		finally{
			System.out.println("This will always be executed.");
		}
	}
}
```

Output:

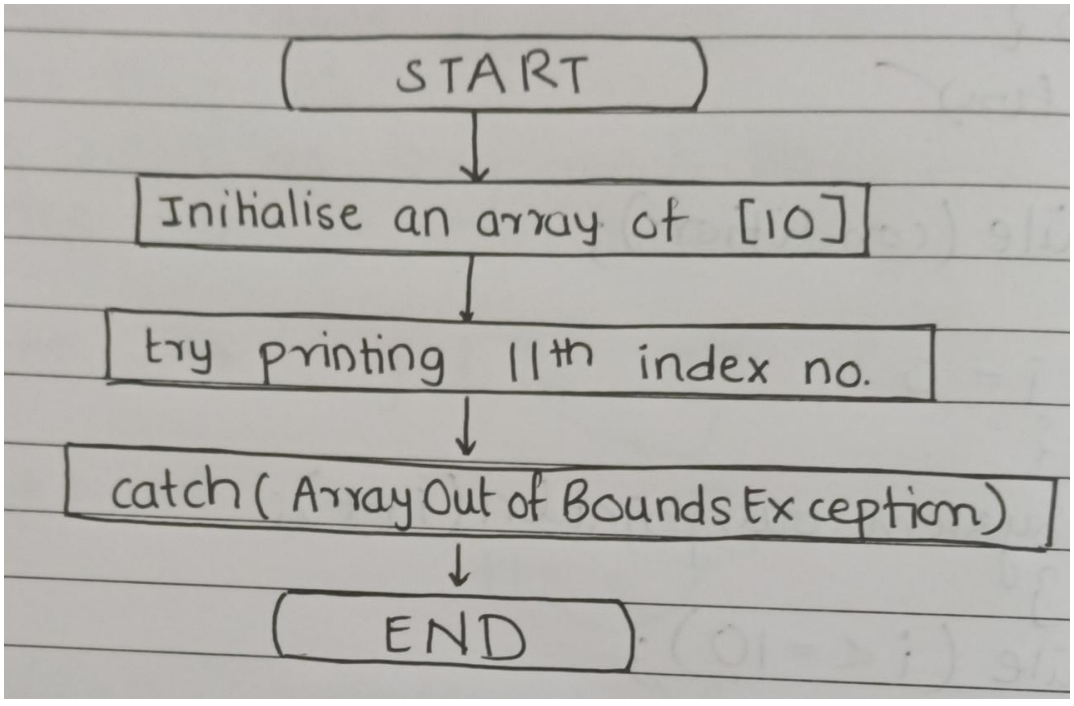
```
Cannot be divided by Zero.
This will always be executed.
```

Java Program 2: Use try and catch to give Array exception.

Algorithm:

1. Start the program.
2. Initialise an array of size 10.
3. Now in the try block write the code, that print element of the array of index 11 using for loop.
4. In catch(ArrayIndexOutOfBoundsException e), print the Exception.
5. End the program.

Flowchart:



Code:

```
package exception;

public class Program2 {
	public static void main(String[] args) {
		int n[] = new int[10];
		for(int i=0; i<=n.length; i++){
		    System.out.println(n[11]);
		}
	}
}
```

Output:

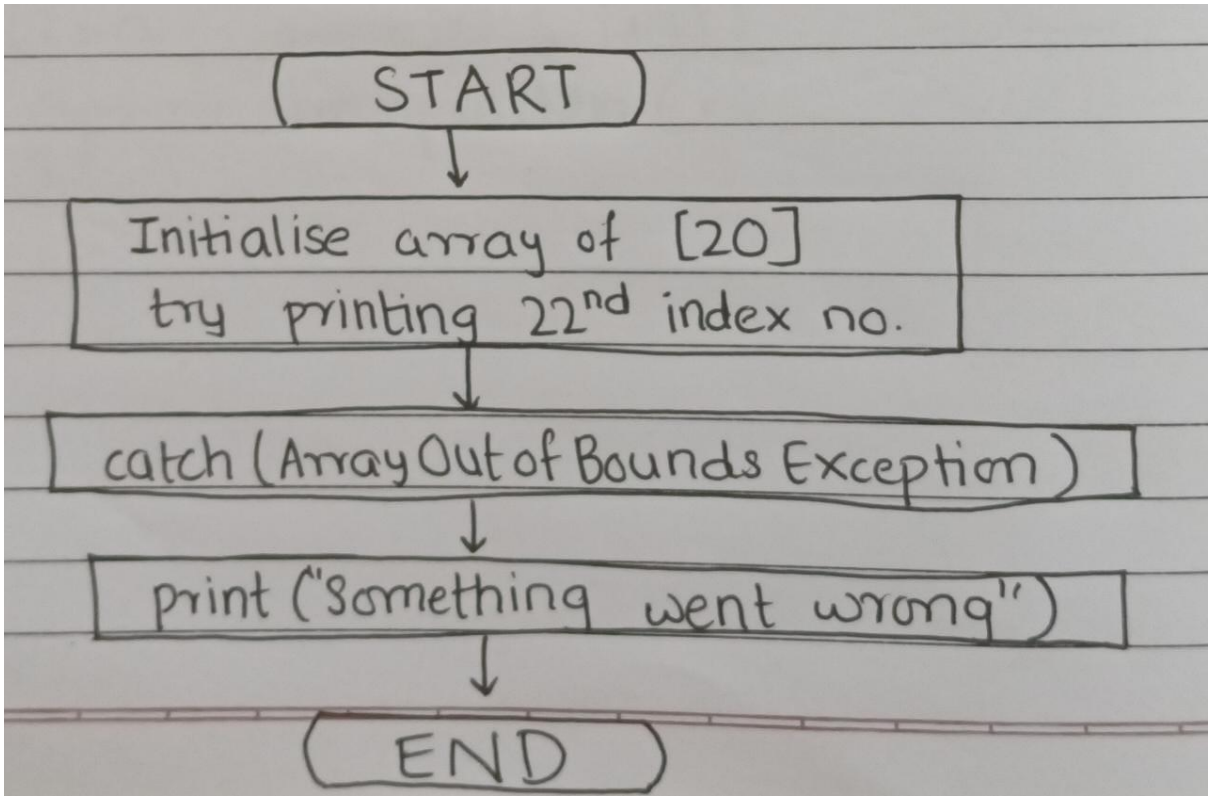
```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 11
	at exception.Program1.main(Program1.java:8)
```

Java Program 3: Use try and catch to give Array exception with statement "Something went wrong."

Algorithm:

1. Initialise an array of size 20.
2. Now in the try block write the code, that print element of the array of index 22 using for loop.
3. In catch(ArrayIndexOutOfBoundsException e), print the Exception.
4. Then in finally print the statement that something went wrong.

Flowchart:



Code:

```
package exception;

public class Program3 {
	public static void main(String[] args) {
		int n[] = new int[20];

		try{
			for(int i=0; i<=n.length; i++){
				System.out.println(n[22]);
			}
		}
		catch(ArrayIndexOutOfBoundsException e){
			System.out.println("Something went wrong.");
		}
		finally{
			System.out.println("This will always get executed.");
		}
	}
}
```

Output:

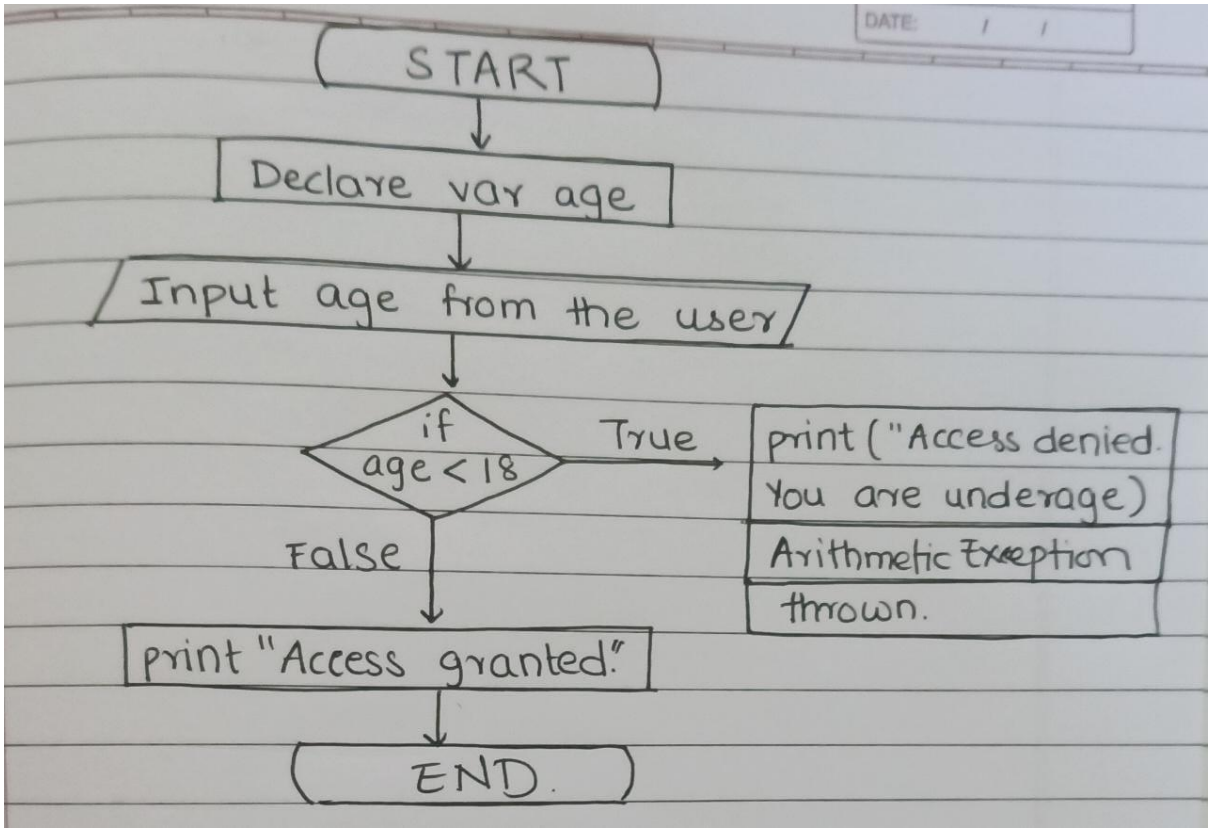
```
Something went wrong.
This will always get executed
```

Java Program 4: Accept age from user and give the access for voting accordingly using catch and throw keyword.

Algorithm:

1. Create a class Program4 and write if else statement using throw new Exception in main method.
2. Now accept the age from user and create object of the class AgeLimit and call the method validAge.

Flowchart:



Code:

```
package exception;
import java.util.Scanner;

public class Program4 {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		int age;

		System.out.println("Enter your age : ");
		age = sc.nextInt();

		if (age <= 18) {
			throw new ArithmeticException("Access Denied. You are
underage.");
			}

		else {
			System.out.println("Access granted. Eligible for
Voting.");
			}
```

Output:

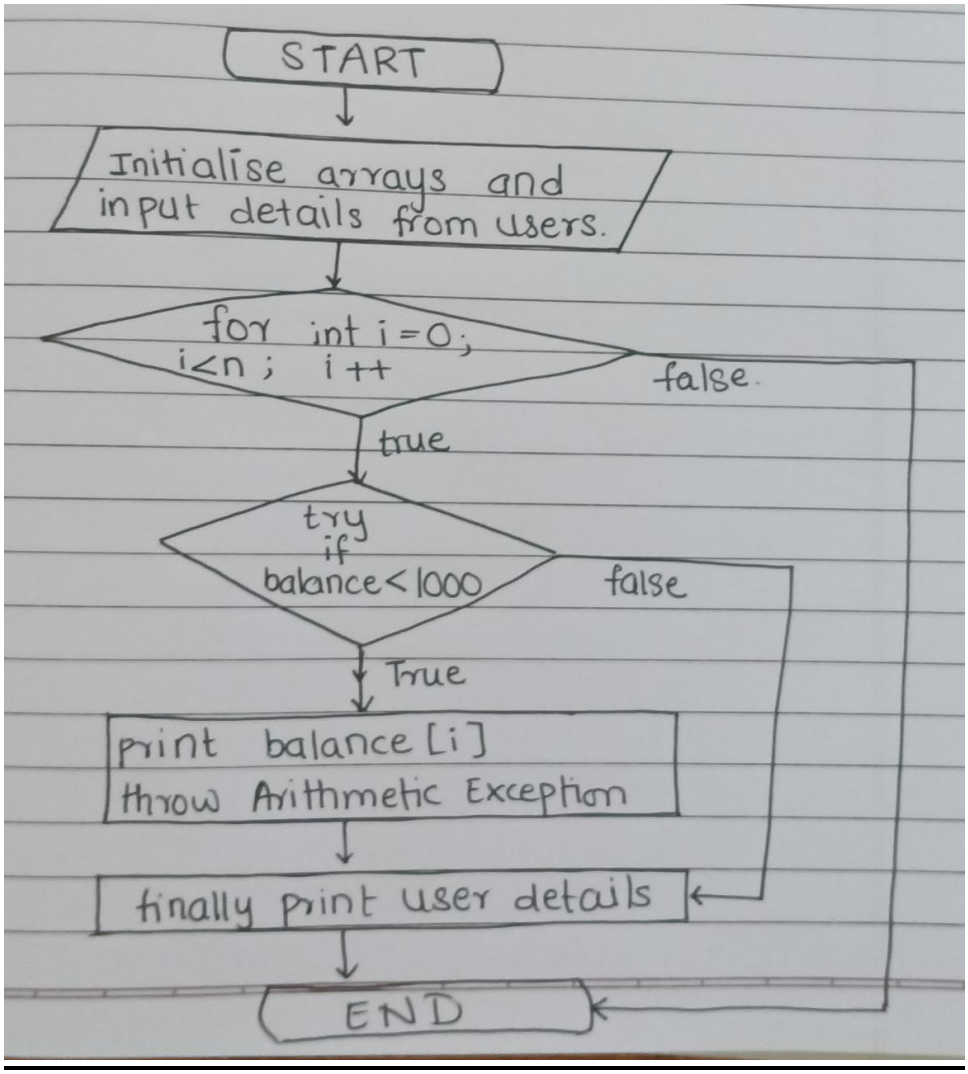
```
Enter your age :
55
Access granted. Eligible for Voting.


Enter your age :
16
Exception in thread "main" java.lang.ArithmeticException: Access Denied.
You are underage.
	at exception.Program1.main(Program1.java:13)
```

Java Program 5: Accept number of customers, name, account number and balance from the user and throw exception when the balance<1000.

Algorithm:

1. Start the program.
2. Declare three arrays for name, account number and balance.
3. Accept number of customers from the user.
4. Then input all the details in for loop. Then in for loop, write the try block containing throw new Exception.
5. Then in finally print the details of the customer.
6. End the program.

Flowchart:



Code:

```
package customer;

import java.util.Scanner;

public class Banking {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        int n;
        int[] balance = new int[10];
        int[] accnum = new int[10];
        String[] name = new String[10];

        System.out.println("Enter number of accounts: ");
        n = sc.nextInt();

        for (int i = 0; i < n; i++) {
            System.out.println("Enter account number: ");
            accnum[i] = sc.nextInt();
            System.out.println("Enter name: ");
            name[i] = sc.next();
            System.out.println("Enter balance: ");
            balance[i] = sc.nextInt();
        }
        sc.close();

        for (int i = 0; i < n; i++) {
            try {
                if (balance[i] < 1000) {
                    System.out.println("Balance: " + balance[i]);
                    throw new ArithmeticException("Balance is not
sufficient");
                }
                else {
                  System.out.println("Account number: " + accnum[i]);
                    System.out.println("Name: " + name[i]);
                    System.out.println("Balance: " + balance[i]);
                }
            }

            catch (ArithmeticException e) {
                System.out.println(e.getMessage());
            }

            finally {
                System.out.println("Thank you and have a nice day!");
            }
        }
    }
}
```

Output:

```
Enter number of accounts:
3
Enter account number:
101
Enter name:
Innie
Enter balance:
12900
Enter account number:
202
Enter name:
Minnie
Enter balance:
990
Enter account number:
303
Enter name:
Mo
Enter balance:
1001

Account number: 101
Name: Innie
Balance: 12900
Thank you and have a nice day!
Balance: 990
Balance is not sufficient
Thank you and have a nice day!
Account number: 303
Name: Mo
Balance: 1001
Thank you and have a nice day!
```