Q1. Write a java program to handle Exception using try, catch, finally blockwhile reading input from command line and store to integer array ?

ANS:

```java
import java.util.Scanner;
public class ExceptionHandlingExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[] numbers = new int[5];

        try {
            System.out.println("Enter five integers:");
            for (int i = 0; i < 5; i++) {
                numbers[i] = Integer.parseInt(scanner.nextLine());
            }
        } catch (NumberFormatException e) {
            System.out.println("Invalid input! Please enter integers only.");
        } finally {
            scanner.close();
        }

        System.out.println("Numbers entered:");
        for (int number : numbers) {
            System.out.println(number);
        }
    }
}
```

OUTPUT:

```
Enter five integers:
1
1
2
1
1
Numbers entered:
1
1
2
1
1
```

Q2. Write a java program for Method level exception handling, for writing data to file using objects ?

ANS:

```java
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;

class Employee implements java.io.Serializable {
    private String name;
    private int age;
```

```java
    public Employee(String name, int age) {
        this.name = name;
        this.age = age;
    }

    @Override
    public String toString() {
        return "Employee [name=" + name + ", age=" + age + "]";
    }
}

public class ExceptionHandlingExample {
    public static void writeToFile(Employee employee, String fileName) throws
IOException {
        try (FileOutputStream fos = new FileOutputStream(fileName);
             ObjectOutputStream oos = new ObjectOutputStream(fos)) {
            oos.writeObject(employee);
            System.out.println("Data has been written to the file successfully.");
        }
    }

    public static void main(String[] args) {
        Employee employee = new Employee("John Doe", 30);
        String fileName = "employee.dat";

        try {
            writeToFile(employee, fileName);
        } catch (IOException e) {
            System.out.println("An error occurred while writing data to the file: "
+ e.getMessage());
        }
    }
}
```

OUTPUT:

Data has been written to the file successfully.

Q3.Write a java program to illustrate, the user can check error conditions and call
the catch block?

ANS:

```java
public class ErrorConditionExample {
    public static void main(String[] args) {
        try {
            int result = divide(10, 0);
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Error: Division by zero is not allowed.");
        }
    }

    public static int divide(int dividend, int divisor) {
        if (divisor == 0) {
            throw new ArithmeticException("Division by zero");
        }
```

```
        return dividend / divisor;
    }
}
```

OUTPUT:

Error: Division by zero is not allowed.

Q4. Write a java program to illustrate IO exception?

ANS:

```java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class IOExceptionExample {
    public static void main(String[] args) {
        BufferedReader reader = null;
        try {
            // Open the file
            reader = new BufferedReader(new FileReader("LAB5.txt"));

            // Read and display the contents of the file
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file: " +
e.getMessage());
        } finally {
            try {
                if (reader != null) {
                    reader.close();
                }
            } catch (IOException e) {
                System.out.println("An error occurred while closing the file: " +
e.getMessage());
            }
        }
    }
}
```


OUTPUT:

An error occurred while reading the file: LAB5.txt (No such file or directory)