



**Vidyavardhini's College of Engineering and Technology**

**Department of Artificial Intelligence & Data Science**

Experiment No. 9
Implement a program on Exception handling.
Date of Performance:
Date of Submission:



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

**Aim:** Implement a program on Exception handling.

**Objective:** To able handle exceptions occurred and handle them using appropriate keyword

### Theory:

The Exception Handling in Java is one of the powerful mechanisms to handle the runtime errors so that the normal flow of the application can be maintained.

Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc.

Java Exception Keywords

Java provides five keywords that are used to handle the exception. The following table describes each.

Keyword	Description
try	The "try" keyword is used to specify a block where we should place an exception code. It means we can't use try block alone. The try block must be followed by either catch or finally.
catch	The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.
finally	The "finally" block is used to execute the necessary code of the program. It is executed whether an exception is handled or not.
throw	The "throw" keyword is used to throw an exception.
throws	The "throws" keyword is used to declare exceptions. It specifies that there may occur an exception in the method. It doesn't throw an exception. It is always used with method signatures.

### Code:

```
public class JavaExceptionExample {  
    public static void main(String args[]) {  
        try {  
  
            int data = 100 / 0;  
        } catch (ArithmeticException e) {
```



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

```
        System.out.println(e);
    }
    System.out.println("rest of the code...");
}
}
```

### OUTPUT:

```
C:\Users\Sharvari A Bhondekar\OneDrive\Desktop\JAVA PROGRAMS\Exp 9>javac JavaExceptionExample.java
C:\Users\Sharvari A Bhondekar\OneDrive\Desktop\JAVA PROGRAMS\Exp 9>java JavaExceptionExample.java
java.lang.ArithmeticException: / by zero
rest of the code...
```

### Conclusion:

Comment on how exceptions are handled in JAVA.

In Java, **exceptions** are handled using the try-catch mechanism. Code that might throw an exception is placed in the **try block**, and if an exception occurs, it is caught and handled by the corresponding **catch block**. This prevents the program from crashing. Java also has the **finally block** to execute code regardless of whether an exception occurred, and you can use **throw** and **throws** to manually trigger or declare exceptions.