



Experiment No. 9
Implement a program on Exceptional Handling
Date of Performance:
Date of Submission:

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

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.

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



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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 signature.

```
public class JavaExceptionExample{  
  
    public static void main(String args[]){  
  
        try{  
  
            //code that may raise exception  
  
            int data=100/0;  
  
  
        }catch(ArithmeticException e){System.out.println(e);}  
  
        //rest code of the program  
  
        System.out.println("rest of the  
code...");  
  
    }  
  
}
```

Output:

```
Exception in thread main java.lang.ArithmeticException:/ by zero  
rest of the code...
```

Code:

```
import java.io.*; class
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

```
ExceptionHandling { public static
void main (String[] args) { int a=5;
int b=0; try{
System.out.println(a/b);
}
catch(ArithmeticException e){
e.printStackTrace();
}
}
}
```

Conclusion:

Exceptions are handled using try-catch blocks. When a potentially exception-causing code is executed, it is placed within a try block. If an exception occurs within the try block, it is caught and handled by the corresponding catch block. This allows for graceful error handling and prevents the program from crashing. Additionally, Java provides the option to specify multiple catch blocks to handle different types of exceptions. This allows for more fine-grained exception handling and enables developers to take appropriate actions based on the specific exception that occurred. Finally, Java also supports the use of finally blocks, which are executed regardless of whether an exception occurred or not. This is useful for cleaning up resources or performing necessary actions before the program exits. Overall, exception handling in Java helps to ensure robust and reliable code execution by gracefully handling unexpected errors.