Experiment No. 9
Implement a program on Exception 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

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 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:

```
public class JavaException


{
 public static void main(String args[])
        {
   try        {
        int data=250/0;
        }
        catch(ArithmeticException e){System.out.println(e);
        }
  System.out.println("rest of the code...");
}
}
```

Conclusion:

Comment on how exceptions are handled in JAVA.

Answer: In Java, exceptions are managed through a structured exception-handling mechanism. When an exceptional event occurs, it throws an exception, which can be caught and handled using `try-catch` blocks. This enables developers to gracefully handle errors and exceptions, preventing program crashes. Exception classes are organized into a hierarchy, with the base class `Throwable`. This hierarchy allows you to catch specific exceptions or handle more general ones. Additionally, you can use the `finally` block for code that should execute regardless of whether an exception is thrown. Java also supports custom exception classes,

letting you create and throw exceptions specific to your application's needs, enhancing error reporting and debugging. Overall, Java's exception handling ensures robust and reliable code by separating error handling from normal program flow.