

Java

Assignment 2.2



Prepared by:

Name of Student : Chaitanya Dalvi

Roll No: 19

Batch: 2023-27

Dept. of CSE

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 1

Title: WAP to demonstrate the functionality of all the methods of String class.

Code:

```
import java.util.Arrays;

public class StringMain {
    public static void main(String[] args) {
        String str = "Hello World";

        System.out.println("Index of 'World': " +
str.indexOf("World"));
        System.out.println("Index of 'World' from index 5: " +
str.indexOf("World", 5));
        System.out.println("Last index of 'o': " +
str.lastIndexOf('o'));
        System.out.println("Substring from index 6: " +
str.substring(6));
        System.out.println("Substring from index 6 to 8: " +
str.substring(6, 8));
        System.out.println("Contains 'World': " +
str.contains("World"));
        System.out.println("Concat '!!!': " + str.concat("!!!"));
        System.out.println("Equals 'Hello World': " +
str.equals("Hello World"));
        System.out.println("Equals ignore case 'hello world': " +
str.equalsIgnoreCase("hello world"));
        System.out.println("Is empty: " + str.isEmpty());
        System.out.println("To lower case: " + str.toLowerCase());
        System.out.println("To upper case: " + str.toUpperCase());
        System.out.println("Compare to 'Hello World': " +
str.compareTo("Hello World"));
        System.out.println("Compare to ignore case 'hello world': "
+ str.compareToIgnoreCase("hello world"));
        System.out.println("Trim: " + str.trim());
        System.out.println("Replace 'o' with '0': " +
str.replace('o', '0'));
        System.out.println("To char array: " +
Arrays.toString(str.toCharArray()));
    }
}
```

```

        System.out.println("Starts with 'Hello': " +
str.startsWith("Hello"));
        System.out.println("Ends with 'World': " +
str.endsWith("World"));
        System.out.println("Join '!!!': " + String.join("!!!",
"Hello", "World"));
        System.out.println("Get bytes: " +
Arrays.toString(str.getBytes()));

        char[] destination = new char[5];
        str.getChars(0, 5, destination, 0);
        System.out.println("Get chars: " +
Arrays.toString(destination));
    }
}

```

Output: (screenshot)

```

/Users/chaitanyadalvi/.zshrc:1: /opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/usr/local/mysql-9.0.1-macos-14-arm64 not found
chaitanyadalvi@chaitanyas-Air assignment2.2 % cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/"
  && javac StringMain.java
  && java StringMain
Index of 'World': 6
Index of 'World' from index 5: 6
Last index of 'o': 7
Substring from index 6: World
Substring from index 6 to 8: Wo
Contains 'World': true
Concat '!!!': Hello World!!
Equals 'Hello World': true
Equals ignore case 'hello world': true
Is empty: false
To lower case: hello world
To upper case: HELLO WORLD
Compare to 'Hello World': 0
Compare to ignore case 'hello world': 0
Trim: Hello World
Replace 'o' with '0': Hello W0rld
To char array: [H, e, l, l, o, , W, o, r, l, d]
Starts with 'Hello': true
Ends with 'World': true
Join '!!!': Hello!!World
Get bytes: [72, 101, 108, 108, 111, 32, 87, 111, 114, 108, 100]
Get chars: [H, e, l, l, o]
chaitanyadalvi@chaitanyas-Air assignment2.2 % 

```

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 2

Title:WAP to demonstrate the functionality of all the methods of StringBuilder class.

```
public class StringBuilderMain {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder("Hello World");

        System.out.println("Append '!!!': " + sb.append("!!!"));
        System.out.println("Insert '!!!' at index 5: " +
sb.insert(5, "!!!"));
        System.out.println("Insert '!!!' at index 5: " +
sb.insert(5, "!!!".toCharArray()));
        System.out.println("Insert '!!!' at index 5: " +
sb.insert(5, "!!!".toCharArray(), 0, 2));
        System.out.println("Insert '!!!' at index 5: " +
sb.insert(5, "!!!".subSequence(0, 2)));
        System.out.println("Insert '!!!' at index 5: " +
sb.insert(5, "!!!".subSequence(0, 2), 0, 2));
        System.out.println("Insert '!!!' at index 5: " +
sb.insert(5, new Object()));
        System.out.println("Replace 'o' with '0': " + sb.replace(4,
5, "0"));
        System.out.println("Delete from index 4 to 5: " +
sb.delete(4, 5));
        System.out.println("Reverse: " + sb.reverse());
        System.out.println("Capacity: " + sb.capacity());
        sb.ensureCapacity(20);
        System.out.println("Ensure capacity: " + sb.capacity());
        System.out.println("Char at index 5: " + sb.charAt(5));
        System.out.println("Length: " + sb.length());
        System.out.println("Substring from index 6: " +
sb.substring(6));
        System.out.println("Substring from index 6 to 8: " +
sb.substring(6, 8));
        System.out.println("Compare to 'Hello World': " +
sb.compareTo(new StringBuilder("Hello World")));
        char[] destination = new char[5];
        sb.getChars(0, 5, destination, 0);
        System.out.println("Get chars: " + new
String(destination));
    }
}
```

```

        System.out.println("Delete char at index 5: " +
sb.deleteCharAt(5));
        System.out.println("Last index of 'o': " +
sb.lastIndexOf("o"));
        System.out.println("Last index of 'o' from index 5: " +
sb.lastIndexOf("o", 5));
        System.out.println("Index of 'llo': " + sb.indexOf("llo"));
        System.out.println("Index of 'llo' from index 5: " +
sb.indexOf("llo", 5));
        sb.setCharAt(5, 'o');
        System.out.println("To string: " + sb.toString());
    }
}

```

Output: (screenshot)

```

cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac StringBuilderMain.java && java StringBuilderMain
/Users/chaitanyadalvi/.zshrc:1: /opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/usr/local/mysql-9.0.1-macos-14-arm64 not found
● chaitanyadalvi@chaitanyas-Air assignment2.2 % cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac String
gBuilderMain.java && java StringBuilderMain
Append '!!!': Hello World!!
Insert '!!!' at index 5: Hello!! World!!
Insert '!!!' at index 5: Hello!!!! World!!
Insert '!!!' at index 5: Hello!!!!!! World!!
Insert '!!!' at index 5: Hello!!!!!!! World!!
Insert '!!!' at index 5: Hello!!!!!!!! World!!
Insert '!!!' at index 5: Hello!!!!!!!!!! World!!
Insert '!!!' at index 5: Hello!!!!!!!!!! World!!
Replace 'o' with '0': Hellojava.lang.Object@63e31ee!!!!!!!!!! World!!
Delete from index 4 to 5: Hellojava.lang.Object@63e31ee!!!!!!!!!! World!!
Reverse: !!dlroW !!!!!!!!!!!ee13e36@tcejb0.gnal.avajlleH
Capacity: 56
Ensure capacity: 56
Char at index 5: o
Length: 46
Substring from index 6: W !!!!!!!!!!!ee13e36@tcejb0.gnal.avajlleH
Substring from index 6 to 8: W
Compare to 'Hello World': -39
Get chars: !!dlr
Delete char at index 5: !!dlrW !!!!!!!!!!!ee13e36@tcejb0.gnal.avajlleH
Last index of 'o': -1
Last index of 'o' from index 5: -1
Index of 'llo': -1
Index of 'llo' from index 5: -1
To string: !!dlro !!!!!!!!!!!ee13e36@tcejb0.gnal.avajlleH
○ chaitanyadalvi@chaitanyas-Air assignment2.2 %

```

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 3

Title: WAP to demonstrate the functionality of the following methods of **StringBuffer** class.

```
public class StringBufferMain {
    public static void main(String[] args) {
        StringBuffer sb = new StringBuffer("Hello World");

        System.out.println("Capacity: " + sb.capacity());
        System.out.println("Char at index 5: " + sb.charAt(5));
        System.out.println("Delete from index 4 to 5: " +
sb.delete(4, 5));
        System.out.println("Delete char at index 5: " +
sb.deleteCharAt(5));
        char[] destination = new char[5];
        sb.getChars(0, 5, destination, 0);
        System.out.println("Get chars: " + new
String(destination));
        System.out.println("Index of 'llo': " + sb.indexOf("llo"));
        System.out.println("Index of 'llo' from index 5: " +
sb.indexOf("llo", 5));
        System.out.println("Last index of 'o': " +
sb.lastIndexOf("o"));
        System.out.println("Last index of 'o' from index 5: " +
sb.lastIndexOf("o", 5));
        System.out.println("Length: " + sb.length());
        System.out.println("Replace from index 4 to 5 with '0': " +
sb.replace(4, 5, "0"));
        System.out.println("Reverse: " + sb.reverse());
        sb.setCharAt(5, 'o');
        System.out.println("Set char at index 5 to 'o': " + sb);
        System.out.println("Substring from index 6: " +
sb.substring(6));
        System.out.println("Substring from index 6 to 8: " +
sb.substring(6, 8));
        System.out.println("To string: " + sb.toString());
    }
}
```

Output: (screenshot)

```
44     System.out.println("Substring from index 6: " + sb.substring(start:6));
45     System.out.println("Substring from index 6 to 8: " + sb.substring(start:6, end:8));
46     System.out.println("To string: " + sb.toString());
47 }
48 }
49
```

OUTPUT TERMINAL 1 PORTS POSTMAN CONSOLE

▼ TERMINAL

```
cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac StringBufferMain.java && java StringBufferMain
/Users/chaitanyadalvi/.zshrc:1: /opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/usr/local/mysql-9.0.1-macos-14-arm64 not found
● chaitanyadalvi@chaitanyas-Air assignment2.2 % cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac StringBufferMain.java && java StringBufferMain
Capacity: 27
Char at index 5:
Delete from index 4 to 5: Hell World
Delete char at index 5: Hell orld
Get chars: Hell
Index of 'llo': -1
Index of 'llo' from index 5: -1
Last index of 'o': 5
Last index of 'o' from index 5: 5
Length: 9
Replace from index 4 to 5 with '0': Hell0orld
Reverse: dlro0lleH
Set char at index 5 to 'o': dlro0leH
Substring from index 6: leH
Substring from index 6 to 8: le
To string: dlro0leH
○ chaitanyadalvi@chaitanyas-Air assignment2.2 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 4

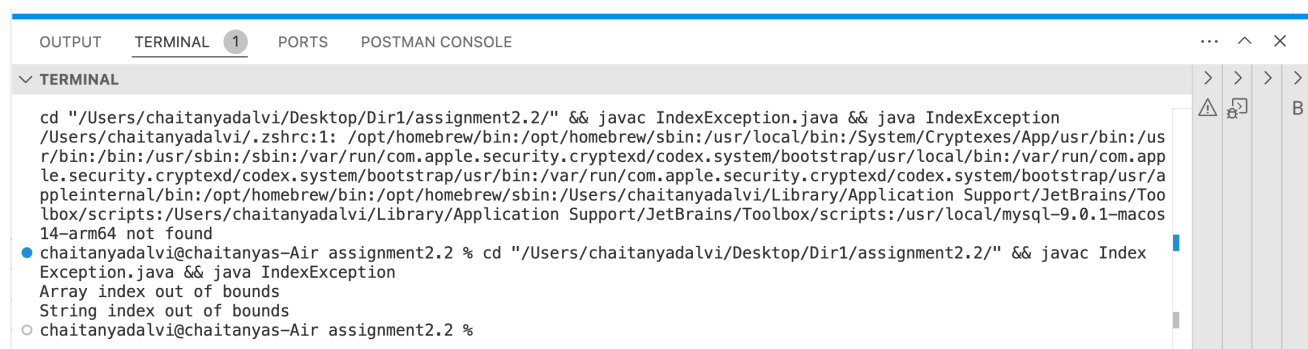
Title: WAP to demonstrate the how to handle `ArrayIndexOutOfBoundsException`, `StringIndexOutOfBoundsException` in a program using multiple catch clause.

Code:

```
public class IndexException {
    public static void main(String[] args) {
        try {
            int[] arr = new int[5];
            System.out.println(arr[5]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Array index out of bounds");
        }

        try {
            String str = "Hello";
            System.out.println(str.charAt(5));
        } catch (StringIndexOutOfBoundsException e) {
            System.out.println("String index out of bounds");
        }
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac IndexException.java && java IndexException
/Users/chaitanyadalvi/.zshrc:1: /opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/usr/local/mysql-9.0.1-macos14-arm64 not found
● chaitanyadalvi@chaitanyas-Air assignment2.2 % cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac IndexException.java && java IndexException
Array index out of bounds
String index out of bounds
○ chaitanyadalvi@chaitanyas-Air assignment2.2 %
```


Name of Student: Chaitanya Dalvi

Roll Number: 19

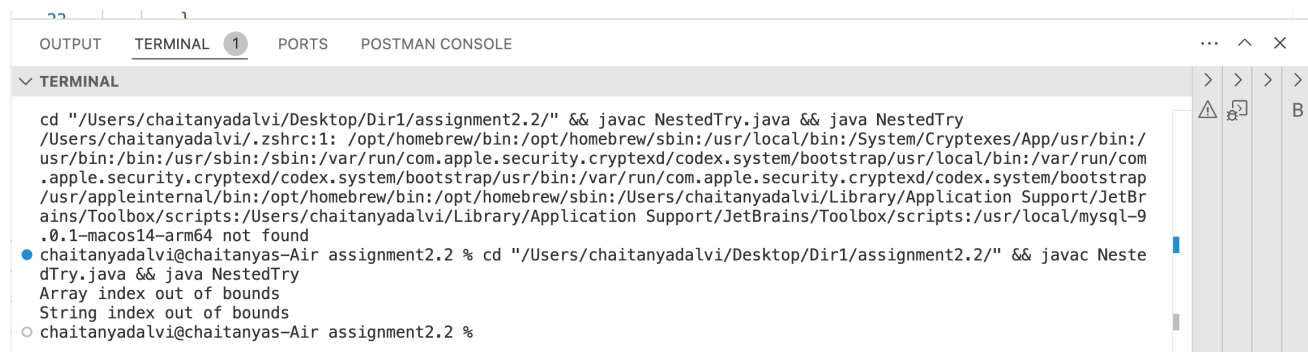
Experiment No: 5

Title:WAP to demonstrate nested try catch statements.**Code:**

```
public class NestedTry {
    public static void main(String[] args) {
        try {
            try {
                int[] arr = new int[5];
                System.out.println(arr[5]);
            } catch (ArrayIndexOutOfBoundsException e) {
                System.out.println("Array index out of bounds");
            }

            try {
                String str = "Hello";
                System.out.println(str.charAt(5));
            } catch (StringIndexOutOfBoundsException e) {
                System.out.println("String index out of bounds");
            }
        } catch (Exception e) {
            System.out.println("Exception caught");
        }
    }
}
```

Output: (screenshot)



The screenshot shows a terminal window with the following output:

```
cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac NestedTry.java && java NestedTry
/Users/chaitanyadalvi/.zshrc:1: /opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/usr/local/mysql-9.0.1-macos14-arm64 not found
● chaitanyadalvi@chaitanyas-Air assignment2.2 % cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac NestedTry.java && java NestedTry
Array index out of bounds
String index out of bounds
○ chaitanyadalvi@chaitanyas-Air assignment2.2 %
```

Name of Student: Chaitanya Dalvi

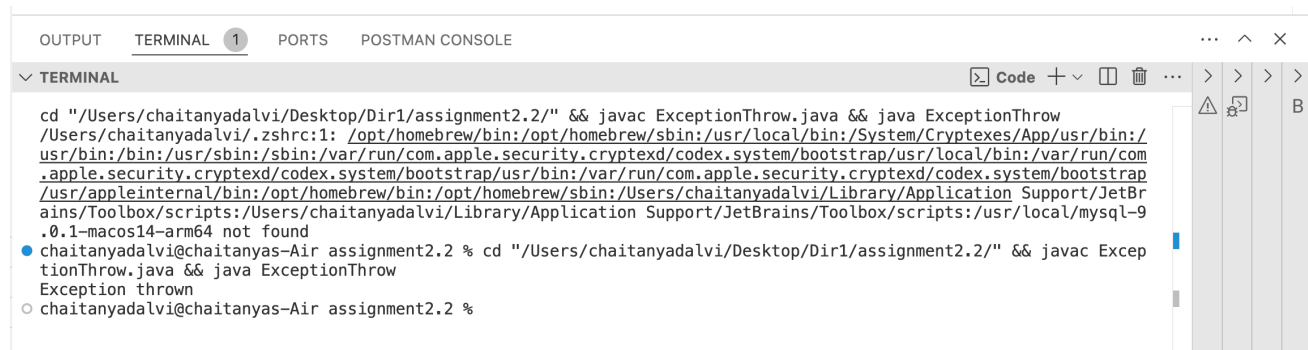
Roll Number: 19

Experiment No: 6

Title: WAP to demonstrate application of throw in exception handling.**Code:**

```
public class ExceptionThrow {  
    public static void main(String[] args) {  
        try {  
            throw new Exception("Exception thrown");  
        } catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac ExceptionThrow.java && java ExceptionThrow  
/Users/chaitanyadalvi/.zshrc:1: /opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/usr/local/mysql-9.0.1-macos14-arm64 not found  
● chaitanyadalvi@chaitanyas-Air assignment2.2 % cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac ExceptionThrow.java && java ExceptionThrow  
Exception thrown  
○ chaitanyadalvi@chaitanyas-Air assignment2.2 %
```

Name of Student: Chaitanya Dalvi


Roll Number: 19

Experiment No: 7

Title: WAP to demonstrate application of throws in exception handling. **Code:**

```
public class ExceptionThrows {  
    public static void main(String[] args) throws Exception {  
        throw new Exception("Exception thrown");  
    }  
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac ExceptionThrows.java && java ExceptionThrows  
/Users/chaitanyadalvi/.zshrc:1: /opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/usr/local/mysql-9.0.1-macos14-arm64 not found  
chaitanyadalvi@chaitanyas-Air assignment2.2 % cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac ExceptionThrows.java && java ExceptionThrows  
Exception in thread "main" java.lang.Exception: Exception thrown  
    at ExceptionThrows.main(ExceptionThrows.java:6)  
chaitanyadalvi@chaitanyas-Air assignment2.2 %
```

Name of Student: Chaitanya Dalvi

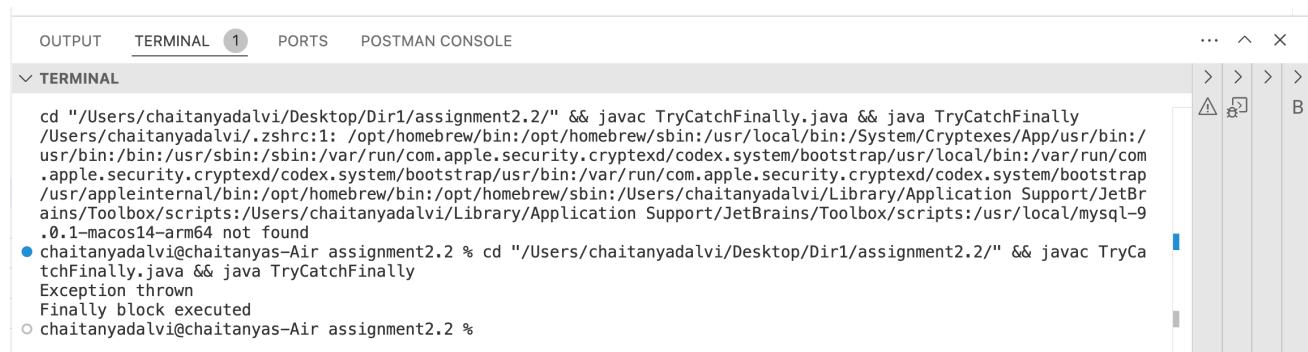
Roll Number: 19

Experiment No: 8

Title: WAP to demonstrate application of try, catch, finally in exception handling. Demonstrate the sequence in which these clauses will be executed

```
public class TryCatchFinally {
    public static void main(String[] args) {
        try {
            throw new Exception("Exception thrown");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        } finally {
            System.out.println("Finally block executed");
        }
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac TryCatchFinally.java && java TryCatchFinally
/Users/chaitanyadalvi/.zshrc:1: /opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/usr/local/mysql-9.0.1-macos14-arm64 not found
● chaitanyadalvi@chaitanyas-Air assignment2.2 % cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac TryCatchFinally.java && java TryCatchFinally
Exception thrown
Finally block executed
○ chaitanyadalvi@chaitanyas-Air assignment2.2 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 9

Title: WAP to find whether a number is prime or not.

```
class InsufficientFundsException extends Exception {
    InsufficientFundsException(String message) {
        super(message);
    }
}

class Account {
    int accountNo;
    double balance;

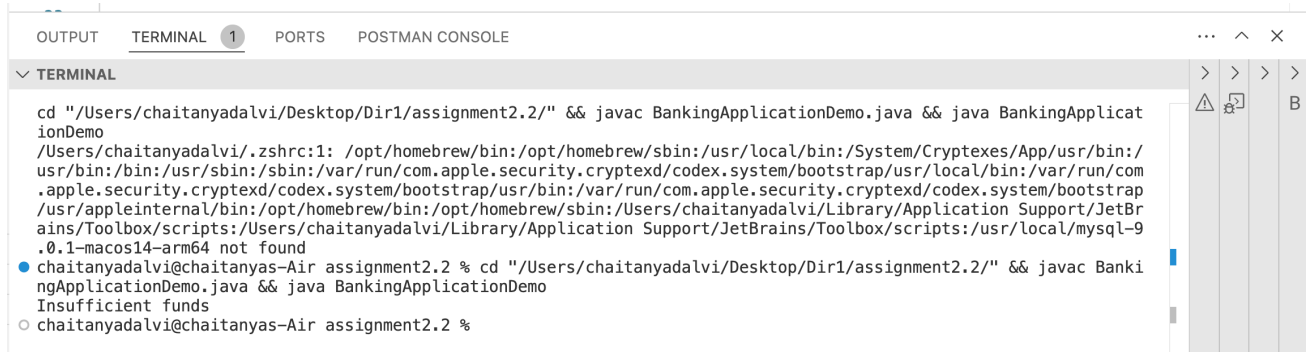
    void deposit(double amt) {
        balance += amt;
    }

    void withdraw(double amt) throws InsufficientFundsException {
        if (amt > balance) {
            throw new InsufficientFundsException("Insufficient
funds");
        }
        balance -= amt;
    }
}

public class BankingApplicationDemo {
    public static void main(String[] args) {
        Account account = new Account();
        account.accountNo = 123;
        account.balance = 1000;

        try {
            account.withdraw(2000);
        } catch (InsufficientFundsException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac BankingApplicationDemo.java && java BankingApplicationDemo
/Users/chaitanyadalvi/.zshrc:1: /opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/usr/local/mysql-9.0.1-macos14-arm64 not found
● chaitanyadalvi@chaitanyas-Air assignment2.2 % cd "/Users/chaitanyadalvi/Desktop/Dir1/assignment2.2/" && javac BankingApplicationDemo.java && java BankingApplicationDemo
Insufficient funds
○ chaitanyadalvi@chaitanyas-Air assignment2.2 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 10

Title: Write a Java Program find out Students Grades using Switch Case

Code:

```
import java.util.Scanner;

public class StudentGrades {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the student's score (0-100): ");
        int score = scanner.nextInt();

        char grade;

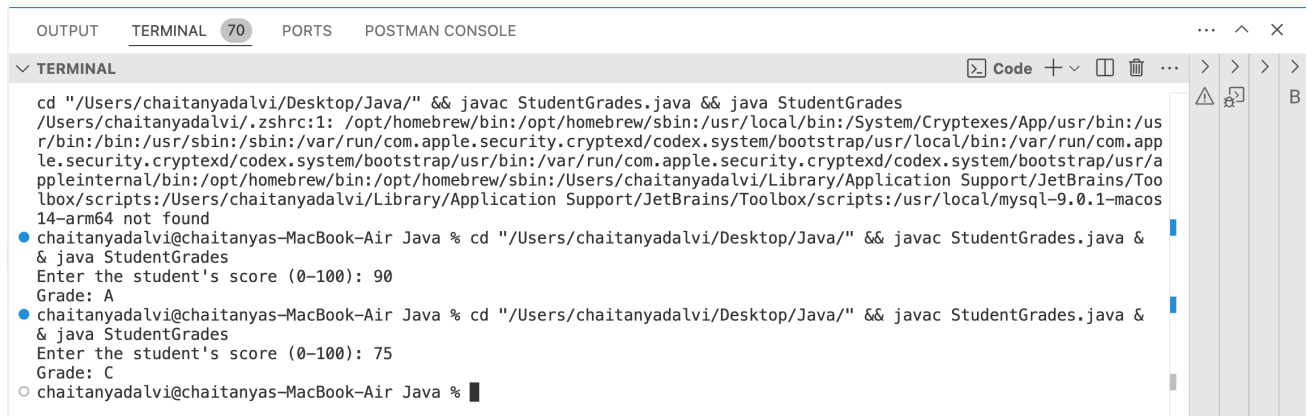
        if (score >= 90) {
            grade = 'A';
        } else if (score >= 80) {
            grade = 'B';
        } else if (score >= 70) {
            grade = 'C';
        } else if (score >= 60) {
            grade = 'D';
        } else {
            grade = 'F';
        }

        switch (grade) {
            case 'A':
                System.out.println("Grade: A");
                break;
            case 'B':
                System.out.println("Grade: B");
                break;
            case 'C':
                System.out.println("Grade: C");
                break;
            case 'D':
                System.out.println("Grade: D");
                break;
            case 'F':
                System.out.println("Grade: F");
        }
    }
}
```

```
        break;
    }

    scanner.close();
}
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/" && javac StudentGrades.java && java StudentGrades
/Users/chaitanyadalvi/.zshrc:1: /opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/System/Cryptexes/App/usr/bin:/usr/bin:/bin:/usr/sbin:/sbin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/local/bin:/var/run/com.apple.security.cryptexd/codex.system/bootstrap/usr/appleinternal/bin:/opt/homebrew/bin:/opt/homebrew/sbin:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/Users/chaitanyadalvi/Library/Application Support/JetBrains/Toolbox/scripts:/usr/local/mysql-9.0.1-macos-14-arm64 not found
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/" && javac StudentGrades.java && java StudentGrades
Enter the student's score (0-100): 90
Grade: A
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/" && javac StudentGrades.java && java StudentGrades
Enter the student's score (0-100): 75
Grade: C
○ chaitanyadalvi@chaitanyas-MacBook-Air Java %
```


Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 11

Title: Differentiate between String, StringBuilder, StringBuffer.

	String	StringBuilder	StringBuffer
Mutability	Immutable (Cannot be changed)	Mutable (Can be changed)	Mutable (Can be changed)
Thread Safety	Not thread-safe	Not thread-safe	Thread-safe (Synchronized)
Performance	Slower for modifications (new object created each time)	Faster for modifications	Slower than StringBuilder (due to thread safety)
Use Case	Best for small, unchanging strings	Best for single-threaded, frequent modifications	Best for multi-threaded environments where thread safety is needed
Synchronized	No	No	Yes
Memory Efficiency	Uses more memory for changes (creates new object)	More efficient for modifications	Less efficient than StringBuilder

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 12

Title: Differentiate between abstract class and Interface

Feature	Abstract Class	Interface
Definition	A class that can have both abstract and concrete methods	A collection of abstract methods (prior to Java 8)
Multiple Inheritance	Cannot support multiple inheritance	Supports multiple inheritance
Methods	Can have abstract and concrete (non-abstract) methods	Only abstract methods (Java 7 and below). Java 8+ allows default and static methods
Variables	Can have instance variables (fields)	Can only have <code>public static final</code> constants (fields)
Constructor	Can have constructors	Cannot have constructors
Access Modifiers	Methods can have any access modifier (public, protected, private)	All methods are <code>public</code> by default
Use Case	Used when classes share common behavior	Used to define a contract for classes to implement
Default Methods	Can have regular methods (non-abstract)	From Java 8, can have <code>default</code> methods and <code>static</code> methods
Multiple Inheritance of Behavior	No, a class can extend only one abstract class	Yes, a class can implement multiple interfaces

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 13

Title: Differentiate between Method overloading and method overriding

Feature	Method Overloading	Method Overriding
Definition	Multiple methods with the same name but different parameters (type, number, or order).	Redefining a method of a parent class in the subclass with the same signature (name, parameters, and return type).
Class	Happens within the same class .	Happens between parent and child classes .
Parameters	Requires different parameter lists.	Parameters must be exactly the same as the parent method.
Return Type	Can have different return types .	Must have the same return type (or a covariant return type).
Method Signature	Method name is the same, but parameters differ .	Method name, parameters, and return type must be identical .
Purpose	Used to increase readability by defining similar tasks with variations in input.	Used to provide specific behavior in a subclass that's different from the parent class.
Inheritance	Not required . Overloading occurs in a single class.	Requires inheritance . A subclass overrides a method of the parent class.
Access Modifiers	Can have any access modifier .	Access modifier must be same or more accessible than the parent method.

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 14

Title: Differentiate between Compile time polymorphism and run time polymorphism.

Aspect	Compile Time Polymorphism	Run Time Polymorphism
Definition	Polymorphism is determined during compile time.	Polymorphism is determined during runtime.
Also Known As	Static Polymorphism / Early Binding	Dynamic Polymorphism / Late Binding
How it's Achieved	Method overloading (same method name, different parameters).	Method overriding (same method name, same parameters but in subclass).
When Decided	Decided by the compiler during code compilation.	Decided by the JVM during program execution.
Performance	Faster, as everything is resolved at compile time.	Slower, because it is resolved at runtime.
Flexibility	Less flexible (fixed at compile time).	More flexible (resolved at runtime).
Example	Multiple methods with the same name but different arguments (method overloading).	A subclass overrides a method from the parent class and it's called using the parent reference (method overriding).
Inheritance Required	Not necessary for method overloading.	Requires inheritance for method overriding.
Code Example	<pre>java class A { void add(int a, int b) { } void add(double a, double b) { } }</pre>	<pre>java class Animal { void sound() { } } class Dog extends Animal { void sound() { } }</pre>

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 15

Title: Differentiate between checked Exceptions and unchecked exceptions.

Aspect	Checked Exceptions	Unchecked Exceptions
Definition	Exceptions that are checked at compile time .	Exceptions that are checked at runtime (during execution).
Handling Requirement	Must be handled using <code>try-catch</code> or declared with <code>throws</code> in the method signature.	Not mandatory to handle or declare explicitly.
When Occurs	Generally occur due to external issues (like file not found, network issues, etc.).	Generally occur due to programming errors (like logic flaws).
Inheritance	Subclasses of <code>Exception</code> but not subclasses of <code>RuntimeException</code> .	Subclasses of <code>RuntimeException</code> .
Example Scenario	External system issues: reading files, database connections, etc.	Coding mistakes: dividing by zero, accessing null objects, etc.
Examples	File not found, Class not found, SQL issues.	Null reference, array index out of bounds, dividing by zero.
Performance	Slightly slower due to compile-time checks.	Faster since no compile-time checking is needed.

Checked Exception	Description
<code>IOException</code>	Occurs during input-output operations, like file handling.
<code>SQLException</code>	Occurs when there is an error interacting with a database.
<code>FileNotFoundException</code>	Happens when a file that is expected is not found.
<code>ClassNotFoundException</code>	Thrown when trying to load a class that is not found.
<code>InterruptedException</code>	Occurs when a thread is interrupted during execution.

Unchecked Exception	Description
<code>NullPointerException</code>	Occurs when trying to access an object with a <code>null</code> reference.
<code>ArrayIndexOutOfBoundsException</code>	Happens when trying to access an invalid index in an array.
<code>ArithmeticException</code>	Thrown when illegal arithmetic operations are performed, like division by zero.
<code>IllegalArgumentException</code>	Happens when a method receives an inappropriate argument.
<code>NumberFormatException</code>	Thrown when trying to convert a string into a number and the format is invalid.