Checked Error:

1.IOException

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
public class IOException {
public static void main(String[] args) throws FileNotFoundException {
BufferedReader br = new BufferedReader(new FileReader("Exceptions.txt"));
String line = br.readLine(); //error:Unhandled exception type
IOExceptionJava(16777384) String java.io.BufferedReader.readLine() throws
IOException
}
}
```

2.FileNotFoundException

```
import java.io.File;
import java.io.FileReader;
public class FileNotFound {
public static void main(String[] args) {
File file = new File("Exceptions.txt");
FileReader reader = new FileReader(file);
}
}
```

error : Unhandled exception type FileNotFoundExceptionJava(16777384)
java.io.FileReader.FileReader(File file) throws FileNotFoundException

3.SQLException

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
public class SQLException {
public static void main(String[] args) {
String jdbcUrl = "jdbc:mysql://localhost:3000/users";
```

```
String username = "root";
String password = "password";
String createTableSQL = "CREATE TABLE IF NOT EXISTS users (" +
"id INT PRIMARY KEY," +
"name VARCHAR(255)," +
"email VARCHAR(255)";            //Invalid table format
try (Connection connection = DriverManager.getConnection(jdbcUrl, username, password);
Statement statement = connection.createStatement()) {
statement.executeUpdate(createTableSQL);      // Attempt to execute the SQL statement to create the table
System.out.println("Table 'users' created successfully.");
} catch (SQLException e) {
System.out.println("SQLException occurred: " + e.getMessage());
}
}
}
```

4.NoSuchMethodException

```
import java.lang.reflect.Method;
public class NoSuchMethodException {
public static void main(String[] args) {
try {
Method method = ExampleClass.class.getMethod("doSomething");
method.invoke(new ExampleClass());
} catch (NoSuchMethodException e) {
System.out.println("NoSuchMethodException occurred: " + e.getMessage());
} catch (Exception e) {
System.out.println("An unexpected exception occurred: " + e.getMessage());
}
}
}

class ExampleClass {
public void doSomething() {                //if doSomething is missing, it invokes NoSuchMethod Exception
```

```
System.out.println("Doing something...");
}
}
```

5.ClassNotFoundException

```
public class ClassnotFound {
public static void main(String[] args) throws ClassNotFoundException {
Class.forName("hello"); //error:Exception in thread "main"
java.lang.ClassNotFoundException: hello
}
}
```

Unchecked Error

1.NullPointerException

```
class NullPointerExample {
public static void main(String[] args) {
try {
String str = null;
System.out.println(str.length());          //error:NullPointerException occurred:
Cannot invoke "String.length()" because "<local1>" is null
} catch (NullPointerException error) {
System.out.println("NullPointerException occurred: " + error.getMessage());
}
}
}
```

2.ArithmeticException

```
class ArthmeticExample {
public static void main(String[] args) {
try {
int result = 1 / 0;          //error:ArithmeticException occurred: / by zero
} catch (ArithmeticException error) {
System.out.println("ArithmeticException occurred: " + error.getMessage());
}
```

```
}
}
```

3.ArrayIndexOutofBoundsException

```
class ArrayOutOBExample {
public static void main(String[] args) {
try {
int[] arr = new int[5];
System.out.println(arr[10]);        //error:Index 10 out of bounds for length 5
ArrayIndexOutOfBoundsException
} catch (ArrayIndexOutOfBoundsException error) {
System.out.println("ArrayIndexOutOfBoundsException occurred: " + error.getMessage());
}
}
}
```

4.UnsupportedOperationException

```
public class UnsupportedOperationExample {
public static void main(String[] args) {
List<String> immutableList = List.of("apple", "banana", "cherry");
try {
immutableList.add("date");        //UnsupportedOperationException occurred: Unsupported operation: List is immutable
} catch (UnsupportedOperationException e) {
System.out.println("UnsupportedOperationException occurred: " + e.getMessage());
}
}
}
```

5.IllegalArugmentException

```
class BMI {
public static void main(String[] args) {
try {
int height = -5;
if (height < 0) {
throw new IllegalArgumentException("height cannot be negative");        //error:IllegalArgumentException
```

```
}
} catch (IllegalArgumentException error) {
System.out.println("IllegalArgumentException occurred: " + error.getMessage());
}
}
```