

with private data member
income, string city
has a default constructor
of class Hotel:
input values for data
members of data member

member functions of class
getcity(), getroomtype()
follows.

City Not Valid
Room type not valid

check if above conditions
are satisfied then

file.

Devanshu Surana
PC-23, 1032210755
Batch C1

OOCCJ Lab Assignment 4B

Aim: Write a Java program to showcase the use of File I/O
Exception Handling in Java.

Part - 1 (File I/O)

A) Read the data from console (Student Rno, name, marks of
5 subjects [calculate Grade]) by InputStreamReader and Buffer-
Reader and write the same to another file using OutputStream
Writer and BufferedWriter.

Objectives:

- To study I/O stream classes.
- To study Java abstract class.

Theory:

Abstraction is a process of hiding the implementation details
and showing only functionality to the user. Another way, it
shows only essential things to the user and hides the internal
details.

for ex: sending SMS where you type the text and send
the message.

It can be achieved with either abstract class or interfaces.

Java I/O is used to process the input and produce the
output. The java.io package contains all classes required
for input and output operation.

Stream is a sequence of data. It is composed of

Input Stream:

Java application uses an input stream to read data from a source, it may be a file, an array, peripheral device or socket.

Output Stream:

Java application uses an output stream to write data to a destination, it may be a file, an array, peripheral device or socket.

Algorithm:

step 1: start

step 2: Create object of FileWriter

step 3: Pass this object as a parameter to the BufferedWriter class.

step 4: Declare variables

step 5: Take Input

step 6: Write the input value onto the file.

step 7: close the file.

step 8: open the txt file to ensure everything is properly.

Step 9: End.

Input: Data of students

Output: file displays all records.

Conclusion: Thus, we have successfully implemented i/o stream classes.

FAQ's

1) Abstract class can have abstract methods while concrete class can have concrete methods. Abstract class can be declared as abstract and concrete class can be declared as concrete.

Array

String

pipe

file

chara

charA

charF

string

string

pipe

pipe

file

file

is composed of byte

am to read data in
array, peripheral device

am to write data
an array, peripheral

to the Buffered

file.

everything is added

plemented usage

Teacher's Sign.

FAQ's

Abstract class cannot be instantiated using new keyword while concrete class can be instantiated using new keyword. Abstract class may or may not have abstract methods but concrete class can not have abstract methods. Abstract class can not be declared as final class while concrete class can be declared final.

	Character stream	Byte stream
Array	CharArrayReader CharArrayWriter	ByteArrayInputStream ByteArrayOutputStream
String	StringReader StringWriter	StringBuffer StringBufferInputStream
pipe	PipedReader PipedWriter	PipedInputStream PipedOutputStream
file	FileReader FileWriter	FileInputStream FileOutputStream

Part - 2 (Exception)

Problem statement: B) Write a menu driven program for banking system which accept the personal data for Customer (cid, cname, amount). Implement the user-defined / standard exceptions, whenever required to handle the following situations:

1. Amount should be created with minimum amount of 1000rs.
2. For withdrawal of amount, if $\text{wth-amt} > \text{amount}$.
3. cid should be in specific range of 1 to 20.
4. Entered amount should be positive.

Objectives:

Understand the different types of exceptions.

Exception Handling using various Exception classes.

Theory:

The exception handling in Java is one of the powerful mechanism to handle the runtime errors so that the normal flow of the application can be maintained.

In Java, an exception is an event that disturbs the normal flow of program. It is an object which is thrown at runtime. It is a mechanism to handle runtime errors.

Java Exception Keywords:

try, catch, finally, throw, throws.

for eg:

```
public class Java {  
    public static void main() {  
        try {  
            int data = 10010;  
        }  
        catch (Exception e) {  
            s.o.p("Error");  
        }  
    }  
}
```

Teacher's Sign

Types:

- 1) Checked Exception:
classes that directly inherit the throwable class
Runtime Exception & Error are known as Checked Exception
- 2) Unchecked Exception:
classes that inherit the Runtime Exception are known as unchecked Exceptions are checked during runtime
- 3) Error: Errors are irrecoverable.

Pseudo code:

```
class Employee extends Exception {  
    Employee (String s)  
    { super (s); }  
}
```

```
class sample Employee {  
    check id < 0 or > 999  
    throw error;  
}
```

```
main () {  
    myEmpcheck (1111);  
}  
catch (e) {  
    print error;  
}
```

Input: EmpId 1111 is passed
Output: Throw, Error.

Conclusion:

Thus, studied the exception handling concepts in Java

FAQ's

The error indicates trouble that primarily occurs due to the scarcity of system resources. The exceptions are the issues that can occur at runtime and compile time. In Java all the errors are unchecked while exceptions can be both checked and unchecked.

checked

Class main {

public static void main() {

FileReader fr = new FileReader("Path");

BufferedReader br = new BufferedReader(fr)

System.out.println(fr.readLine());

fr.close();

}

}

The above error can be avoided by using try catch block.

unchecked

int x = 10/0;

It throws arithmetic Exception during runtime.

- i) try: Used to specify the exception block.
- ii) catch: used to specify the solution
- iii) finally: The 'finally' keyword has a mandatorily executable code.
- iv) throw: Throws an exception
- v) throws: used to declare an exception

"throw" keyword throws an exception while the "throws" keyword is used to declare an exception.

Teacher's Sign: _____

in Java.

Sign: _____