CORE CONCEPTS

- An exception can occur for many different reasons. Following are some scenarios where an exception occurs. A user has entered an invalid data. A file that needs to be opened cannot be found.
- Checked exceptions A checked exception is an exception that is checked (notified) by the compiler at compilation - time, these are also called as compile time exceptions . These exceptions cannot simply be ignored, the programmer should take care of (handle) these exceptions . Unchecked exceptions An unchecked exception is an exception that occurs at the time of execution. These are also called as Runtime Exceptions .
- These include programming bugs, such as logic errors or improper use of an API. Runtime exceptions are ignore d at the time of compilation. Errors These are not exceptions at all, but problems that arise beyond the control of the user or the programmer. For example, if a stack overflow occurs, an error will arise.
- They are also ignored at the time of compilation. The exception object contains name and description of the exception, and current state of the program where exception has occurred. Now the following procedure will happen. The block of the code is called Exception handler .
- If it finds appropriate handler then it passes the occurred exception to it. Appropriate handler means the type of the exception object thrown matches the type of the exception o bject it can handle. This handler prints the exception information in the following format and terminates program abnormally . Customized Exception Handling: Java exception handling is managed via five keywords: try , catch , th row , throws , and finally .
- Program statements that you think can raise exceptions are contained within a try block. If an exception occurs within the try block , it is thrown. Your code can catch this exception (using catch block) and handle it in some rational manner. To manually throw an exception, use the keyword throw .
- Any code that absolutely must be executed af ter a try block completes is put in a finally block. A try/catch block is placed around the code that might generate an exception. When an exception occurs, that exception occurred is handled by catch block associated with it. Every try block should be immediately followed either by a catch block or finally block.
- A catch statement involves declaring the type of exception you are trying to catch. If an exception occurs in protected code, the catch block (or blocks) that follows the try is checked. println("Something went wrong. "); } } } The output will be: Something went wrong.
- MULTIPLE CATCH BLOCKS A try block can be followed by multiple catch blocks. You can throw an exception, either a newly instantiated one or an exception that you just ca ught, by using the throw keyword. A finally block of code always executes, irrespective of occurrence of an Exception. Using a finally block allows you to run any cleanup - type statements that you want to execute, no matter what happens in the protected code.
- println ("The finally statement is executed"); } } } Output Exception thrown :java. It is not compulsory to have finally clauses when ever a try/catch block is present. The try block cannot be present without either catch clause or finally clause. Any code cannot be present in between the try, catch, finally blocks.
- USER - DEFINED CUSTOM EXCEPTION IN JAVA User Defined Exception or custom exception is creating your own exception class and throws that exception using throw keyword. This can be done by extending the class Exception. But practically, you will require some amount of customizing as per your programming needs. Output String = Custom message BCA - JAVA PRORAMMING YUVAKSHETRA INSTITUT E OF MANAGEMENT STUD IES BCA 2017 ONWARS BATCH Page 8 Java - Files and I/O The java.
- io package contains nearly every class you might ever need to perform input and output (I/O) in Java. All these streams represent an input sour ce and an output destination. io package supports many data such as primitives, object, localized characters, etc. STREAM A stream can be defined as a sequence of data.

- There are two kinds of Streams InPutStream The InputStream is used to read data from a source. OutPutStream The OutputStream is used for writing data to a destination. In Java, 3 streams are created for us automatically. All these streams are attached with the console.
- BCA - JAVA PRORAMMING YUVAKSHETRA INSTITUT E OF MANAGEMENT STUD IES BCA 2017 ONWARS BATCH Page 9 BYTE STREAMS Java byte streams are used to perform input and output of 8 - bit bytes. Though there are many classes related to byte streams but the most frequently used classes are, FileInputStream and FileOutputStream . Following is an example which makes use of these two classes to copy an input file into an output file Example import java . *; public class CopyFile { public static void main (String args []) throws IOException { FileInputStream in = null ; FileOutputStream out = null ; try { in = new FileInputStream ("input.
- txt"); out = new FileOutputStream ("output. txt"); int c ; while ((c = in . write (c); } } finally { if (in ! = null) { BCA - JAVA PRORAMMING YUVAKSHETRA INSTITUT E OF MANAGEMENT STUD IES BCA 2017 ONWARS BATCH Page 10 in .
- close (); } } } } Now let"s have a file input. txt with the following con tent This is test for copy file. txt file with the same content as we have in input. So let"s put the above code in CopyFile.
- java file and do the following $javac CopyFile. java $java CopyFile CHARACTER STREAMS Java Byte streams are used to perform input and output of 8 - bit bytes, whereas Java Character streams are used to perform input and output for 16 - bit unicode. Though there are many classes related t o character streams but the most frequently used classes are, FileReader and FileWriter . We can re - write the above example, which makes the use of these two classes to copy an input file (having unicode characters) into an output file Example import java .
- t xt"); out = new FileWriter ("output. txt"); int c ; while ((c = in . write (c); } } finally { if (in ! close (); } } } } Now let"s have a file input.
- txt with the following content This is test for copy file. txt file with the same content as we have in input. So let"s put the above code in CopyFile. java fi le and do the following $javac CopyFile.
- Following is a simple program, which creates InputStreamReader to read stan dard input stream until the user types a "q" Example import java . println ("Enter characters, "q" to quit. "); char c ; do { c = (char) cin . print (c); } while (c !
- = "q"); } finally { if (cin ! close (); } } } } BCA - JAVA PRORAMMING YUVAKSHETRA INSTITUT E OF MANAGEMENT STUD IES BCA 2017 ONWARS BATCH Page 13 Let"s keep the above code in ReadConsole. This program continues to read and output the same character until we press "q" $javac ReadConsole. java $java ReadConsole Enter characters, "q" to quit.
- 1 1 e e q q READING AND WRITING FILES As described earlier, a stream can be defined as a sequence of data. The InputStream is used to read data from a source and the OutputStream is used for writing data to a destination. Here is a hierarchy of classes to deal with Input and Output streams. The two important streams are FileInputStream and FileOutputStream , which would be discussed in this tutorial.
- BCA - JAVA PRORAMMING YUVAKSHETRA INSTITUT E OF MANAGEMENT STUD IES BCA 2017 ONWARS BATCH Page 14 FileInputStream This stream is used for reading data from the files. Objects can be created using the keyword new and there are several types of constructors available. Returns the next byte of data and - 1 will be returned if it"s the end of the file. length bytes from the input stream into an array.
- Returns the total number of bytes read. If it is the end of the file, - 1 will be returned. 5 public int available() throws IOException{} Gives the number of bytes that c an be read from this file input stream. BCA - JAVA PRORAMMING YUVAKSHETRA INSTITUT E OF MANAGEMENT STUD IES BCA 2017 ONWARS BATCH Page 15 FileOutputStream FileOutputStream is used to create a file and write data into it.

- The stream would create a file, if it doesn"t already exist, before opening it for output. Here are two constructor s which can be used to create a FileOutputStream object. 4 public void write(byte[] w) Writes w. length bytes from the mentioned byte array to the OutputStream.

  Example Following is the example to demonstrate InputStream and OutputStr eam import java .