FILE HANDLING PROGRAMS

1. Java program to create a new file
2. Java program to write content into file using FileOutputStream
3. Java program to read content from file using FileInputStream
4. Java program to write content into file using BufferedWriter
5. Java program to read content from file using BufferedReader
6. Java program to get file size and file path
7. Java program to delete a file
8. Java program to copy files
9. Java program to get the last modification date and time of a file
10. Java program to append text/string in a file
11. Java program to determine number of bytes written to file using DataOutputStream
12. Java program to read text from file from a specified index or skipping byte using FileInputStream
13. Java program to check whether a file is hidden or not
14. Java program to get the size of given file in bytes, kilobytes and megabytes
15. Java program to create directory/folder in particular drive
16. Java program to check whether a file can be read or not
17. Java program to read and print all files from a zip file
18. Java program to get the attributes of a file
19. Java program to get the basic file attributes (specific to DOS)
20. Java program to get the file's owner name
21. Java program to get file creation, last access and last modification time
22. Java program to read content from one file and write it into another file
23. Java program to read a file line by line
24. Java program to traverse all files of a directory/folder
25. Java - Print File Content, Display File using Java Program
26. Java - Copy Content of One File to Another File using Java Program
27. Write a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in byte
28. Write a program to show how to read and write a file.
29. Write a program to create a text file in the path c:\java\abc.txt and check whether that file is exists. Using the command exists(), isDirectory(), isFile(), getName() and getAbsolutePath().

30 .Write a program to rename the given file, after renaming the file delete the renamed file. (Accept the file name using command line arguments.)
31. Write a program to create a directory and check whether the directory is created.
32. Write a program to create a file and write data into it using the methods OutputStream class.
33. Write a program to accept a specified number of characters as input and converts them into uppercase characters.
34. Write a program to get the input from the user and store it into file. Using Reader and Writer file.
****

MultiThreading

1] Write a program to illustrate creation of threads using runnable class.(start method start each of the newly created thread. Inside the run method there is sleep() for suspend the thread for 500 milliseconds).
2] Write a program to create a class MyThread in this class a constructor, call the base class constructor, using super and starts the thread. The run method of the class starts after this. It can be observed that both the main thread and created child thread are executed concurrently.
3] Write a program to get the reference to the current thread by calling currentThread() method.
4] Write a program to create two threads. In this class we have one constructor used to start the thread and run it. Check whether these two threads are run are not.
5] Create a multithreaded program by creating a subclass of Thread and then creating, initializing, and staring two Thread objects from your class. The threads will execute concurrently and display Java is hot, aromatic, and invigorating to the console window.
6] Create a multithreaded program as in the previous exercise by creating the MyThread subclass of Thread. But create threads as objects of the class MyClass, which is not a subclass of Thread. MyClass will implement the runnable interface and objects of MyClass will be executed as threads by passing them as arguments to the Thread constructor
7] Write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

****

EXCEPTION HANDLING

1]Create your user-defined exception class by inheriting the built-in Exception class. Next, create a constructor for your custom exception class. You can do this by either writing a default constructor within the CustomException or, you may create a

parameterized constructor with a string argument.[Text Wrapping Break] This parameterized constructor can be used to store the details of the exception.
The ‘throw’ keyword is used to raise the user-defined exception. We create an object of the user-defined exception class and the throw clause to initiate that object.

2|Write java code to implement exception handling. Use of keywords throw, throws, try, catch, and finally. Checked and unchecked exceptions. Writecode that deals with both checked and unchecked exceptions. Write custom exception classes by extending base exception classes from the API.Code using try-with-resources.

3|Write a program that creates a Calculator class. The class contains two variables of integer type. Design a constructor that accepts two values as parameter and set those values.
Design four methods named Add (), Subtract (), multiply (), Division ( ) for performing addition, subtraction, multiplication and division of two numbers.
For addition and subtraction, two numbers should be positive. If any negative number is entered then throw an exception in respective methods. So design an exception handler (ArithmeticException) in Add () and Subtract () methods respectively to check whether any number is negative or not.
For division and multiplication two numbers should not be zero. If zero is entered for any number then throw an exception in respective methods. So design an exception handler (ArithmeticException) in multiply () and Division () methods respectively to check whether any number is zero or not.
Write a main class and declare four objects of Calculator class. Perform addition (obj1), subtraction (obj2), multiply (obj3) and division (obj4) operations for these objects. If any non integer values are provided as input; then you should throw an exception (NumberFormatException) and display a message that informs the user of the wrong input before exiting.

4|Create an exception class named MyException that extend a base class named Exception
Design a constructor in your class that accepts a string value set it to the super class constructor to display the exception message.
Create a main class named product. Write a method inside the class called productCheck(int weight) that accepts weight of the product. Inside the method, if the weight is less than 100 then throw an exception “Product is invalid” otherwise print the weight of the product.
Inside the main method declare single object of the product class and call the productCheck() method to display the weight of the product

****