**1 . Write the programme to open a text file named input 2, and copy its contents to an output text file output.**

**Program:**

**package** myPackage;

**import** java.io.\*;

**public** **class** File\_Clone {

**public** **static** **void** main(String[] args) {

**try** {

// Create File objects for input and output files

File inputFile = **new** File("C:/Users/shrey/OneDrive/Desktop/Anudip Documentation/New folder/input.txt");

File outputFile = **new** File("C:/Users/shrey/OneDrive/Desktop/Anudip Documentation/New folder/output.txt");

// Initialize BufferedReader to read from the input file

BufferedReader reader = **new** BufferedReader(**new** FileReader(inputFile));

// Initialize BufferedWriter to write to the output file

BufferedWriter writer = **new** BufferedWriter(**new** FileWriter(outputFile));

String line;

// Read each line from the input file and write it to the output file

**while** ((line = reader.readLine()) != **null**) {

writer.write(line);

writer.newLine(); // Write a new line in the output file after each line

}

// Close the BufferedReader and BufferedWriter to release resources

reader.close();

writer.close();

System.***out***.println("File copied successfully."); // Confirmation message

} **catch** (IOException e) {

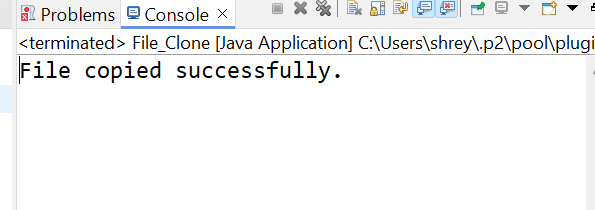
// Handle any IO exceptions that occur

e.printStackTrace();

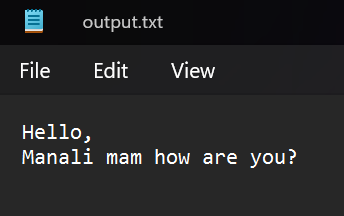
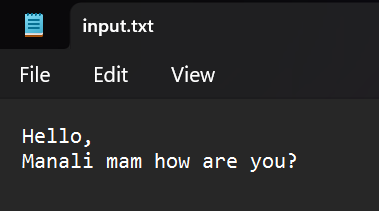
}

}

}

**Output: **

**Input.text Output.text:**

****

1. **Write the programme to show multithreading for the string “multi threads”. Show the resulting output.**

**Program:**

**package** myPackage;

**public** **class** Multithreading\_Demo **extends** Thread {

**private** String message;

// Constructor to initialize the message

**public** Multithreading\_Demo(String message) {

**this**.message = message;

}

// Override the run method to define the behavior of the thread

@Override

**public** **void** run() {

// Print the message (in this case, a single character)

System.***out***.println(message);

}

**public** **static** **void** main(String[] args) {

// The message to be printed using multiple threads

String message = "multi threads";

// Loop through each character in the message string

**for** (**int** i = 0; i < message.length(); i++) {

// Create a new thread for each character

Multithreading\_Demo thread = **new** Multithreading\_Demo(String.*valueOf*(message.charAt(i)));

// Start the thread, which will execute the run method

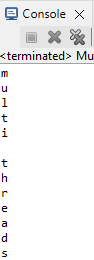
thread.start();

}

}

}

**Output:**

****

1. **Implement a Java program that creates a thread using the Runnable interface. The thread should print numbers from 1 to 10 with a delay of 1 second between each number.**

**Program:**

**package** myPackage;

**public** **class** PrintNumbers\_Demo **implements** Runnable {

@Override

**public** **void** run() {

**try** {

// Loop to print numbers from 1 to 10

**for** (**int** i = 1; i <= 10; i++) {

System.***out***.println(i);

// Pause the thread for 1 second (1000 milliseconds)

Thread.*sleep*(1000);

}

} **catch** (InterruptedException e) {

// Handle the interrupted exception if the thread is interrupted during sleep

e.printStackTrace();

}

}

**public** **static** **void** main(String[] args) {

// Create a new thread with the PrintNumbers\_Demo instance

Thread thread = **new** Thread(**new** PrintNumbers\_Demo());

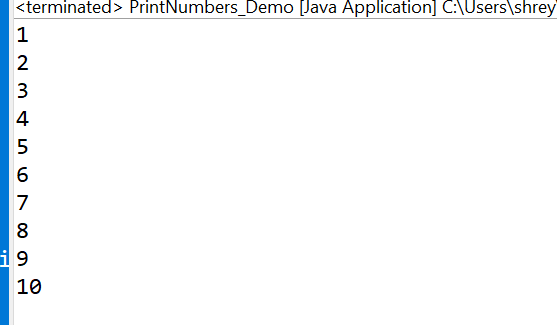
// Start the thread, which will execute the run method

thread.start();

}

}

**Outuput**

****

1. **Write a Java program that creates and starts three threads. Each thread should print its name and count from 1 to 5 with a delay of 500 milliseconds between each count.**

**Program:**

**package** myPackage;

**public** **class** Thread\_Demo **extends** Thread {

**public** Thread\_Demo(String name) {

**super**(name);

}

// Override the run method to define the thread's behavior

@Override

**public** **void** run() {

**try** {

// Loop to print the thread name and count from 1 to 5

**for** (**int** i = 1; i <= 3; i++) {

System.***out***.println(getName() + ": " + i);

// Pause the thread for 500 milliseconds (0.5 seconds) between prints

Thread.*sleep*(500);

}

} **catch** (InterruptedException e) {

// Handle the interrupted exception if the thread is interrupted during sleep

e.printStackTrace();

}

}

**public** **static** **void** main(String[] args) {

Thread t1 = **new** Thread\_Demo("Thread 1");

Thread t2 = **new** Thread\_Demo("Thread 2");

Thread t3 = **new** Thread\_Demo("Thread 3");

t1.start(); // Start the first thread

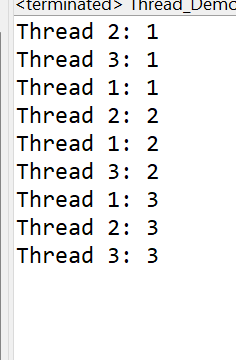
t2.start(); // Start the second thread

t3.start(); // Start the third thread

}

}

**Output:**

****

1. **Create a Java program that demonstrates thread priorities. Create three threads with different priorities and observe the order in which they execute.**

**Program:**

**package** myPackage;

**public** **class** ThreadPriorities\_Demo **extends** Thread {

// Constructor to set the thread name

**public** ThreadPriorities\_Demo(String name) {

**super**(name);

}

// Override the run method to define the behavior of the thread

@Override

**public** **void** run() {

// Print the name of the thread and its priority

System.***out***.println(getName() + " is running with priority " + getPriority());

}

**public** **static** **void** main(String[] args) {

// Create three threads with different names

ThreadPriorities\_Demo t1 = **new** ThreadPriorities\_Demo("Low Priority Thread");

ThreadPriorities\_Demo t2 = **new** ThreadPriorities\_Demo("Medium Priority Thread");

ThreadPriorities\_Demo t3 = **new** ThreadPriorities\_Demo("High Priority Thread");

// Set the priority of each thread

t1.setPriority(Thread.***MIN\_PRIORITY***); // Set priority to 1 (lowest priority)

t2.setPriority(Thread.***NORM\_PRIORITY***); // Set priority to 5 (normal priority)

t3.setPriority(Thread.***MAX\_PRIORITY***); // Set priority to 10 (highest priority)

// Start each thread

t1.start();

t2.start();

t3.start();

}

}

**Output:**

