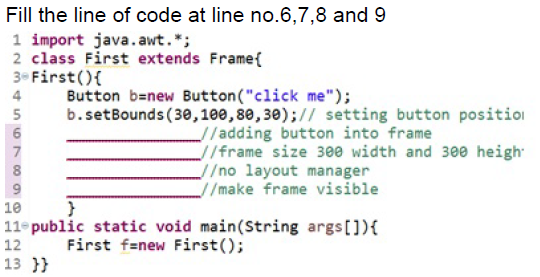
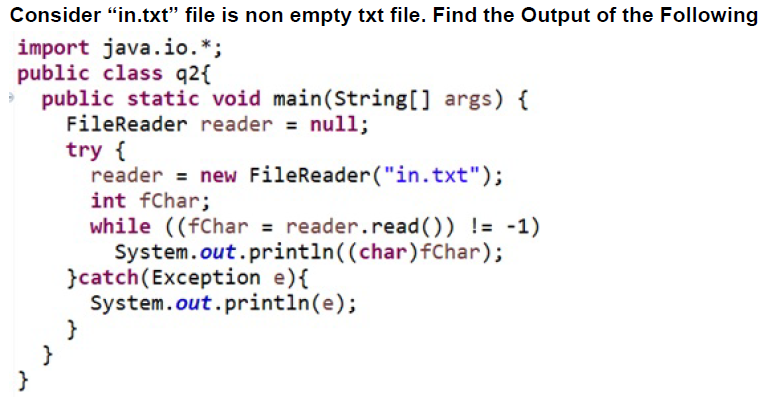
|  |
| --- |
| 1. Which of these packages contains all the classes and methods required for event handling in Java? |

1. Java.applet
2. Java.awr
3. Java.event
4. **Java.awt.event**
5. Complete the code –



1. 6: add(b); 7: setSize (300,300); 8: setLayout(FlowLayout); 9: setVisible(true);
2. **6: add(b); 7: setSize (300,300); 8: setLayout (null); 9: setVisible(true);**
3. 6:add(button); 7: setSize (300,300); 8: setLayout (null); 9: setVisible(true);
4. 6:add(b); 7: setSize (300,300); 8:setLayout (null); 9: setVisible ();
5. Which of the following displays components row-by-row in the order in which they were added to the JFrame?
6. CardLayout
7. **FlowLayout**
8. BorderLayout
9. GridLayout
10. If programmer wants to copy content from File1.txt to File2.txt using Character Stream Class, then which of the following is correct?
11. FileInputStream("File1.txt") and FileOutputStream("File2.txt");
12. **FileReader("File1.txt") and FileWriter("File2.txt");**
13. FileReader("File2.txt") and FileWriter("File1.txt");
14. FileInputStream("File2.txt") and FileOutputStream("File1.txt");



1. Read and Print Content from File ”in.txt” in Single line
2. Error at while ((fChar = reader.read()) != -1)
3. Error at line System.*out*.println((char)fChar);
4. Read and Print Content from File ”in.txt” in separate line
5. **Write the Java program for write a data in abc.txt file.**

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.\*;

import java.util.\*;

class writedata

{

public static void main(String[] args)

{

try {

FileWriter fr = new FileWriter("abc.txt");

Scanner sc=new Scanner(System.in);

System.out.println("eEnter data :");

String str=sc.nextLine();

int i=0;

str += (char)i;

System.out.println(str);

fr.write(str);

fr.close();

System.out.println("File reading and done");

}

catch (IOException e) {

System.out.println(

"There are some IOException");

}

}

}

1. **Write the Java program for read a data from abc.txt file and display on screen.**

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

class GFG

{

public static void main(String[] args)

{

try {

FileReader fr = new FileReader("gfgInput.txt");

String str = "";

int i;

while ((i = fr.read()) != -1) {

str += (char)i;

}

System.out.println(str);

fr.close();

System.out.println("File reading and done");

}

catch (IOException e) {

System.out.println(

"There are some IOException");

}

}

}

1. **Write a program for create a frame and add two labels, two text fields for Student\_ID and Student\_Name and one button for Submit.**
2. **Write a Java program for file handling operations – create a new file, check that file is readable /writable or not ,find the length of a file and ,display the name of file, display absolute path of a file and delete the data from file**.

import java.io.\*;

import java.util.Scanner;

public class practice {

public static void main(String[] args) {

int choice;

Scanner sc = new Scanner(System.in);

System.out.println("Press 1 for Creating a file:");

System.out.println("Press 2 for checking whether file is readable");

System.out.println("Press 3 for checking whether file is Writable");

System.out.println("Press 4 for finding length of file:");

System.out.println("Press 5 for displaying name of file:");

System.out.println("Enter your choice: ");

choice = sc.nextInt();

File f = new File("abc.txt");

switch (choice){

case 1:

try {

if(f.createNewFile()){

System.out.println("File created");

}

else{

System.out.println("File already exists");

}

}catch (Exception e) {

e.printStackTrace();

}

break;

case 2:

if(f.exists()){

System.out.println(f.canRead()+" File is Readable");

}

else{

System.out.println("File is not readable");

}

break;

case 3:

if(f.exists()){

System.out.println(f.canWrite()+" File is Writable");

}

else{

System.out.println("File is not Writable");

}

break;

case 4:

if(f.exists()){

System.out.println("The length of file is "+f.length());

}

break;

case 5:

if(f.exists()){

System.out.println("The name of the file is "+f.getName());

}

break;

}

}

}

**Note: For theory questions write small code wherever it required.**

1. Differences between AWT and Swing.

|  |  |
| --- | --- |
| **AWT** | **SWING** |
| A collection of GUI components | A part of Java Foundation classes (JFC) |
| AWT components are heavyweight | Swing components are lightweight |
| AWT are platform dependent | Swing are platform independent |
| It has less number of components | It has more components |
| AWT does not support pluggable look and feel | Swing supports pluggable look and feel |
| It requires java.awt package | It requires javax.swing package |
| Execution is slower | Execution is faster |
| Components require more memory space | Components does not require much memory space |
| AWT doesn’t follow MVC | Swing follows MVC |

1. Explain File Handling concept.

**Solution:**

* In Java, a **File** is an abstract data type.
* A named location used to store related information is known as a **File**.
* With the help of file class, we can able to work with files
* These File class is inside the java.io package
* The file class can be used by creating an object of the class and then specifying the name of the file
* File Handling is an integral part of any programming language as file handling enables us to store the output of any particular program in a file and allows us to perform certain operations on it.
* In simple words, file handling means reading and writing data to a file
* There are several **File Operations** like **creating a new File, getting information about File, writing into a File, reading from a File** and **deleting a File**.

1. Write about layout manager in Java GUI.

**Solution:**

Layout manager means, arranging a component in a sequential manner. Throughout this layouts components can easily add as per mentioned layout. There are various layouts present like flow layout, border layout, card layout, grid layout, etc.

Flow layout is used to arrange components inline means from top left corner to right.

Border layout is used to arrange component at 5 locations east, west, center, north, south. Only one component can able to add at one location

Card layout is like only one panel is visible at a time

Grid layout is used when we want to add/represent data in the form of rows and columns.

1. **Write about event handling in java GUI.**

Event Handling is the mechanism that controls the event and decides what should happen if an event occurs. This mechanism has a code which is known as an event handler, that is executed when an event occurs.

Java uses the Delegation Event Model to handle the events. This model defines the standard mechanism to generate and handle the events.

The Delegation Event Model has the following key participants.

* **Source** − The source is an object on which the event occurs. Source is responsible for providing information of the occurred event to it's handler. Java provide us with classes for the source object.
* **Listener** − It is also known as event handler. The listener is responsible for generating a response to an event. From the point of view of Java implementation, the listener is also an object. The listener waits till it receives an event. Once the event is received, the listener processes the event and then returns.

The benefit of this approach is that the user interface logic is completely separated from the logic that generates the event. The user interface element is able to delegate the processing of an event to a separate piece of code.

In this model, the listener needs to be registered with the source object so that the listener can receive the event notification. This is an efficient way of handling the event because the event notifications are sent only to those listeners who want to receive them.

1. What are the different classes are available for read and write operation for File. Explain in short.

Ans.

In Java, a **File** is an abstract data type. A named location used to store related information is known as a **File**. There are several **File Operations** like **creating a new File, getting information about File, writing into a File, reading from a File** and **deleting a File**.

Before understanding the File operations, it is required that we should have knowledge of **Stream** and **File methods**.

A series of data is referred to as **a stream**. In [Java](https://www.javatpoint.com/java-tutorial), **Stream** is classified into two types, i.e., **Byte Stream** and **Character Stream**.

**Byte Stream** is mainly involved with byte data. A file handling process with a byte stream is a process in which an input is provided and executed with the byte data.

**Character Stream** is mainly involved with character data. A file handling process with a character stream is a process in which an input is provided and executed with the character data.

The next operation which we can perform on a file is **"writing into a file"**. In order to write data into a file, we will use the **FileWriter** class and its **write()** method together. We need to close the stream using the **close()** method to retrieve the allocated resources.

The next operation which we can perform on a file is **"read from a file"**. In order to write data into a file, we will use the **Scanner** class. Here, we need to close the stream using the **close()** method. We will create an instance of the [Scanner class](https://www.javatpoint.com/Scanner-class) and use the **[hasNextLine()](https://www.javatpoint.com/post/java-scanner-hasnextline-method)**[method](https://www.javatpoint.com/post/java-scanner-hasnextline-method) **[nextLine()](https://www.javatpoint.com/post/java-scanner-nextline-method)**[method](https://www.javatpoint.com/post/java-scanner-nextline-method) to get data from the file.