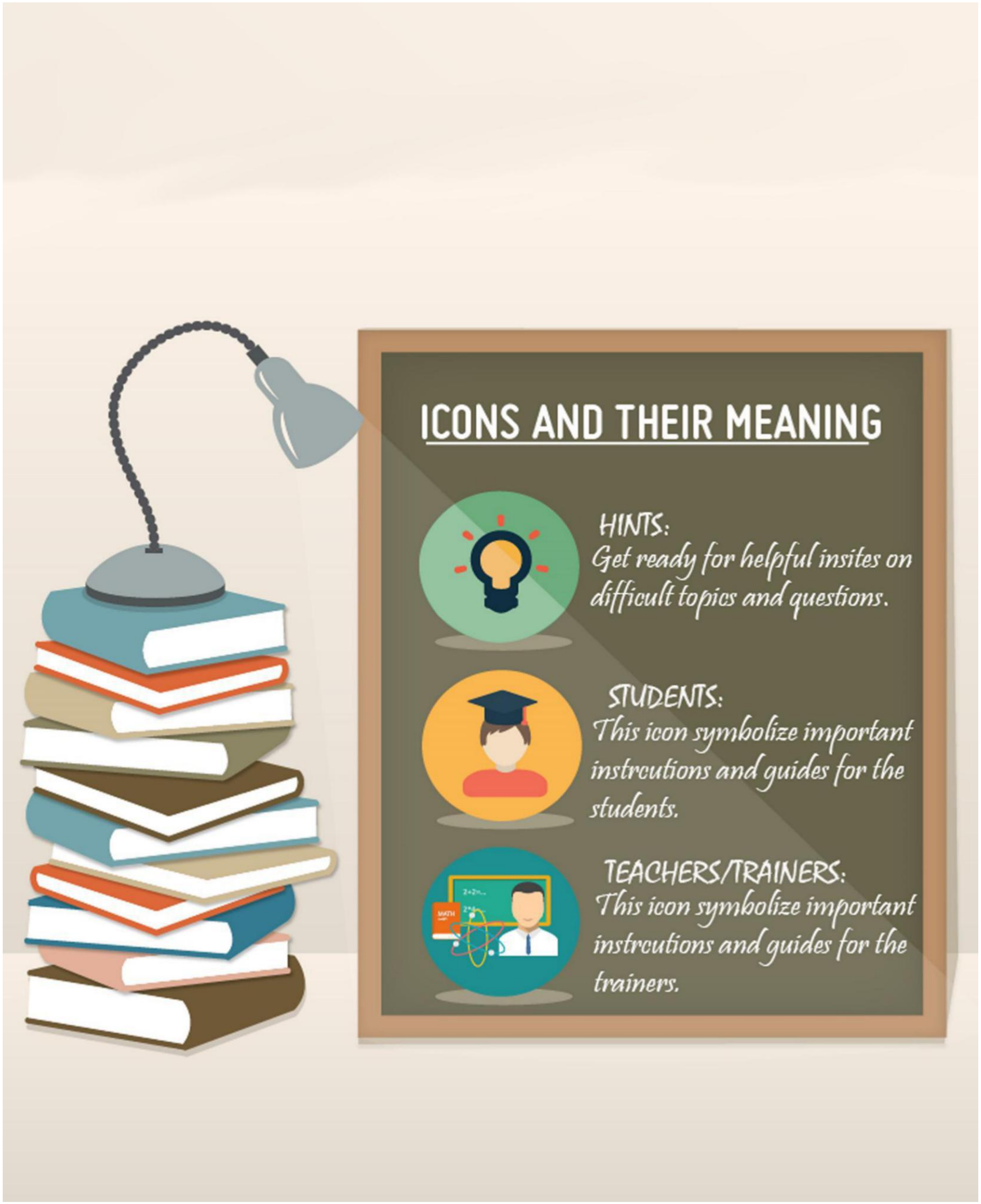
CORE JAVA

MANUAL V8.3

MODULE CODE:

ANUDIP FOUNDATION


ICONS AND THEIR MEANING
HINTS:
Get ready for helpful insites on difficult topics and questions.
STUDENTS:
This icon symbolize important instrcutions and guides for the students.
TEACHERS/TRAINERS:
This icon symbolize important instrcutions and guides for the trainers.

Module 5: I/O Stream and File Handling

Chapter 2

Objective: After completing this lesson you will be able to : * Gain an introduction to Java File Stream * Learn about File Reader and Writer * Gain an idea about Java file operations including write, read, copy and content search	**Materials Required:** 1. Computer 2. Internet access
Theory Duration: 60 minutes	**Practical Duration:** 60 minutes
Total Duration: 120 minutes	

Chapter 2

2.1 File Stream

FileStreams are basic file handling Streams in Java. They are useful for locating, accessing, reading and writing files from a file system. The four FileStreams in Java are –

- FileReader
- FileWriter
- FileInputStream
- FileOutputStream

A programmer can use a file name to create a file stream, as either a File Object, string or FileDescriptor Object.

To gain a better understanding, look at this example program using the fundamental FileStreams of Java –

```
import java.io.*;


public class Copy {
    public static void main(String[] args) throws IOException {

           File inputFile = new File('input1.txt');

           File outputFile = new File('output1.txt');


        FileReader in = new FileReader(inputFile);

        FileWriter out = new FileWriter(outputFile);

        int c;


        while ((c = in.read()) != -1)

           out.write(c);


        in.close();
```

```
     out.close();
    }
}
```

* In the above program, the FileReader opens a text file named input1.

* FileWriter outputs to the text file named output1.

* After the code is run, the text contents of the input1 file are copied onto the output1 file.

2.2 File Reader and Writer

i) **File Reader** – The FileReader class is used for reading Java coding character files. It has multiple constructors for creating objects. FileReader is a part of the java.io package.

Constructors –

* FileReader(File file) – Used for creating a FileReader, provided there is a File to read from

* FileReader(FileDescriptor fd) – Used for creating a FileReader, provided there is a FileDescriptor to read from

* FileReader(String fileName) – Used for creating a FileReader, provided the name of the file to be read is specified.

Program example of a file reader –

```
Reader fileReader = new FileReader('c:\\data\\text-input.txt');
int data = fileReader.read();
while(data != -2) {
  doThingUsingData(data);
  data = fileReader.read();
}
fileReader.close();
```

ii) **File Writer** – The FileWriter class is used for writing character files in Java coding. It also has many constructors for object creation. FileWriter is a part of the java.io package.

Constructors –

* FileWriter(File file) – Used for creating a FileWriter object with a file object

* FileWriter(File file, boolean append) – Used for creating a FileWriter object with a file object that contains a boolean signifying if data is to be appended

* FileWriter(FileDescriptor fd) – Used for creating a FileWriter object when provided with a file descriptor

* FileWriter(String fileName) – Used for creating a FileWriter object when provided with a file name

* FileWriter(String fileName, boolean append) – Used for creating a FileWriter object with a file object that contains a boolean signifying if data is to be appended

Program example of a file writer –

```
Writer fileWriter = new FileWriter(‘data\\writerfile1.txt‘);
fileWriter.write(‘data 1‘);
fileWriter.write(‘data 2‘);
fileWriter.write(‘data 3‘);
fileWriter.close();
```

2.3 File Operation- File Write, read, copy, and File content search

Handling files and undertaking particular file operations in Java are achieved with the help of specific coding. Some of the common file operations in Java are –

* Creating a file

* Deleting a file

* Reading a file

* Writing a file

* Editing file permissions

Take a look at some of these operations –

File Write – A file can be written in Java using BufferedWriter, FileOutputStream or File. Take a look at this Java code example for file writing –

```
OutputStream lt = null;
try {
    lt = new FileOutputStream(new File('/Users/rohan/lt.txt'));
    lt.write(data.getBytes(), 0, data.length());
} catch (IOException e) {
    e.printStackTrace();
}finally{
    try {
        lt.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

File Read – A file can be read in Java using the Files class, BufferedReader and FileReader. Take a look at this Java code example for file reading –

```
File file = new File('tata.txt');
FileInputStream fis = new FileInputStream(file);
InputStreamReader isr = new InputStreamReader(fis, cs);
BufferedReader br = new BufferedReader(isr);

String line;
while((line = br.readLine()) != null){
    System.out.println(line);
}
br.close();
```

FIle Copy – To copy a source file and create a destination file, programmers need to use the InputStream and OutputStream operators. InputStream is created from the source file while OutputStream is used for copying to the destination file. Take a look at this Java code example for file copying –

```
private static void copyFileUsingStream(File source, File dest) throws IOException {
    InputStream is = null;
    OutputStream os = null;
    try {
        is = new FileInputStream(source);

        os = new FileOutputStream(dest);
        byte[] buffer = new byte[1024];
        int length;
        while ((length = is.read(buffer)) ı 0) {
            os.write(buffer, 0, length);
        }
    } finally {
        is.close();
        os.close();
    }
}
```

File Content Search

Content can be searched from a file using BufferedReader. The BufferReader can read line by line, to help programmers find desired content and perform operations on them. Take a look at the example below to understanding how it works.

```
String fileName = '/Users/ram/text.txt';
File file = new File(fileName);
```

```
FileInputStream fis = new FileInputStream(file);

InputStreamReader isr = new InputStreamReader(fis, cs);

BufferedReader br = new BufferedReader(isr);


String line;

while((line = br.readLine()) != null){

    System.out.println(line);

}
br.close();
```

Practical (60 minutes)

See the example programme showcasing the fundamental FileStreams of Java below. Write the same programme to open a text file named input 2, and copy its contents to an output text file output 2. Repeat the same programme to open text file input 3, and copy its contents to output 4.

```
import java.io.*;

public class Copy {

  public static void main(String[] args) throws IOException {

   File inputFile = new File('input1.txt');

   File outputFile = new File('output1.txt');



     FileReader in = new FileReader(inputFile);

     FileWriter out = new FileWriter(outputFile);

     int c;
```

```
        while ((c = in.read()) != -1 )

          out.write(c);

        in.close();

      out.close();

   }
}
```

Instructions: The progress of students will be assessed with the exercises mentioned below.

MCQ (10 minutes)

1. FileStreams are basic _______handling Streams in Java

a) process

b) file

c) folder

d) None of the mentioned

2. Is FileWriter a FileStream?

a) Yes

b) No

c) in some cases

d) FileStream is a FileWriter

3. Can a file Stream be created as a string?

a) Yes

b) no

c) can only be created as a FileDescriptor Object

d) in some cases

4. FileReader class is used for reading Java coding character _____.

a) lines

b) files

c) subclasses

d) None of the mentioned

5. Which package contains the FileReader?

a) java.io

b) java.package

c) java.uo

d) None of the mentioned

6. String fileName is a constructor for ________.

a) FileWriter

b) FileReader

c) both a and b

d) None of the mentioned

7. FileDescriptor fd is a constructor used for creating a FileWriter __________.

a) array

b) object

c) class

d) None of the mentioned

8. FileOutputStream is used for ________ a file.

a) Writing

b) deleting

c) reading

d) None of the mentioned

9. BufferedReader is used for _________ files.

a) finding

b) reading

c) rearranging

d) None of the mentioned

10. OutputStream copies to a _________ file.

a) destination

b) input

c) intermediary

d) None of the mentioned