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 1

Objective: After completing this lesson you will be able to : * Gain an understanding of Java Stream and its types - Byte, Character and Standard Stream * Get an idea about basic control input and output	**Materials Required:** 1. Computer 2. Internet access
Theory Duration: 60 minutes	**Practical Duration:** 60 minutes
Total Duration: 120 minutes	

Chapter 1

1.1 Stream- Byte Stream and Character Stream, Standard Stream

What is a Stream in Java ?

In Java, a stream refers to an object sequence which is capable of supporting multiple methods that can be executed to achieve intended goals. Utilizing java.util.stream enables programmers to access and use the Java Stream API. The Stream package consists of interfaces, classes and enums for carrying out different element operations.

A stream in Java fetches its inputs from Arrays, Collections and I/O channels. It is not a data structure by itself. Streams do not make modifications to the original data structures. Instead, they produce results based on methods that are processed. Code is analyzed by Stream only when required.

This example shows Stream in use for filtering data Collection –

```
import java.util.*;
    import java.util.stream.Collectors;
    class Name{
       int id;
       String name;
       float age;
       public Age(int id, String name, float age) {
          this.id = id;
          this.name = name;
          this.age = age;
       }
    }
    public class JavaStreamExample {
       public static void main(String[] args) {
```

```
        List<Age> ageList = new ArrayList<Age>();
        ageList.add(new Age(1,"Ram",22f));
        ageList.add(new Age(2,"Shyam",24f));
        ageList.add(new Age(3,"Pradip",32f));
        ageList.add(new Age(4,"Dilip",43f));
       List<Float> ageList3 = ageList.stream()
       .filter(a->a.age > 32
       .map(a->a.age)
       .collect(Collectors.toList());
     System.out.println(ageList3);
   }
  }
```

Output:

```
[32]
```

i) **Byte Stream** – Byte Stream is one of the fundamental Stream types in Java. It is capable of managing raw binary data in 8-bit byte segments. A Byte Stream can input and output 8-bit bytes. The most commonly used byte stream classes are FileInputStream and FileOutputStream. This Stream type has many other class types which belong to InputStream and OutputStream. A byte stream is ideal for the processing of binary files or other raw data.

An example program exhibiting the Byte Stream in use –

ByteArrayInputStreamDemo.java

```
import java.io.ByteArrayInputStream;
class IOTest{
   String str = "Bytes and bytes";
   public void readFile(){
         try {
```

```
                byte b[] = str.getBytes( );
                ByteArrayInputStream bais =
                        new ByteArrayInputStream(b);
                int i;
                while((i=bais.read( ))!=-1){
                        System.out.print((char)i);
                }
        } catch (Exception e) {
                e.printStackTrace( );
        }
    }
}
public class ByteArrayInputStreamDemo {
    public static void main(String args[]){
            IOTest obj = new IOTest( );
            obj.readFile( );
  }
}
```

Output:

Bytes and bytes

ii) Character Stream – A Character Stream is useful for performing the input and output of 16-bit Unicode character bytes. It is the other fundamental stream type in Java. Many programmers prefer to use the Character Stream as it is ideal for processing text files. This is because Java uses Unicode conventions to store characters. FileReader and FileWriter are two of the most used classes in the Character Stream.

An example program exhibiting the Character Stream –

```
import java.io.*;
public class ASD
{
```

```
public static void main(String[] args) throws IOException
  {
     FileReader sourceStream = null;
     try
     {
        sourceStream = new FileReader('test1.txt');
        int temp;
        while ((temp = sourceStream.read()) != -1)
            System.out.println((char)temp);
     }
     finally
     {
        if (sourceStream != null)
           sourceStream.close();
     }
  }
}
```

iii) Standard Stream – In Java, a Standard Stream is one that takes input from programmers to display an output on a computer's screen. The three components of the Standard Stream are the standard input, standard output and standard error.

An example program exhibiting Standard Stream in Java –

```
import java.io.*;
public class ReadConsoles {

  public static void main(String args[]) throws IOException {
    InputStreamReader cin = null;
    try {
      cin = new InputStreamReader(System.in);
```

```
  System.out.println("Enter characters, 'k' to quit.");
        char c;
        do {
          c = (char) cin.read();
          System.out.print(c);
        } while(c != 'k');
      }finally {
        if (cin != null) {
          cin.close();
        }
      }
    }
}
```

Output:

```
% javac ReadConsoles.java
% java ReadConsoles Enter characters, 'k' to quit.
1
1
d
d
k
k
```

Basic Console Input and Output

The Console is the operating system window through which users can inetract with console applications and system programs.

* Standard Input – Standard Input, represented as System.in, is the component used for inputting data into programs i.e. usually through a computer keyboard.

* Standard Output - Standard Output, represented as System.out, is the component used for outputting the data from a Java program i.e. usually displayed on a computer screen.

* Standard Error - Standard Error, represented as System.err, is the component used for displaying errors that are produced by a Java program i.e. usually displayed on a computer screen.

Example of Java Console -

```
import java.io.Console;
class StringReadTest{
public static void main(String args[]){
Console c=System.console();
System.out.println('Enter the name: ');
String n=c.readLine();
System.out.println('Hello '+n);
}
}
```

Enter the name: Saurav Lal

Hello Saurav Lal

Practical (60 minutes)

See the example programme for Java Byte Stream below. Write the same programme for the class ByteStream1 and the string "Bytes are real". Show the resulting output. Repeat the same programme for the class BytesStreams and the string "Bytes are bytes".

ByteArrayInputStreamDemo.java

```
import java.io.ByteArrayInputStream;
class IOTest{
  String str = 'Bytes and bytes';
```

```
 public void readFile(){
     try {
      byte b[] = str.getBytes();
        ByteArrayInputStream bais =
                        new ByteArrayInputStream(b);
        int i;
        while((i=bais.read())!=-1){
           System.out.print((char)i);
        }
     } catch (Exception e) {
        e.printStackTrace();
     }
   }
}
public class ByteArrayInputStreamDemo {
   public static void main(String args[]){
        IOTest obj = new IOTest();
             obj.readFile();
 }
}
```

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

MCQ (10 minutes)

1. A stream in Java refers to _____ sequence.

a) an object

b) a subject

c) a list

d) None of the mentioned

2. A stream can support _____ methods.

a) single

b) multiple

c) negative

d) None of the mentioned

3. Which of these does a stream package contain?

a) classes

b) enums

c) interfaces

d) All of the mentioned

4. Which of these does a stream fetch its input from?

a) Collections

b) Arrays

c) both a and b

d) None of the mentioned

5. Streams produce results based on processed _____.

a) classes

b) methods

c) sub-streams

d) None of the mentioned

6. What type of data does the Byte Stream manage?

a) Unicode

b) Alphanumeric

c) raw binary

d) None of the mentioned

7. A Byte Stream can process _____ byte segments.

a) 32-bit

b) 8-bit

c) 16-bit

d) None of the mentioned

8. A Character Stream performs the input of _____ data.

a) Unicode

b) raw binary

c) streaming

d) Both a and b

9. FileReader is a widely used class of the _____ Stream.

a) Byte

b) Character

c) both a and b

d) None of the mentioned

10. Standard input is represented as _____.

a) System.output

b) System.input

c) System.in

d) None of the mentioned