



PROGRAMMING IN JAVA

Assignment 7

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: $10 \times 1 = 10$

QUESTION 1:

Which of these exception is thrown in cases when the file specified for writing is not found?

- a. IOException
- b. FileNotFoundException
- c. FileNotFoundException
- d. FileInputException

Correct Answer:

- c. FileNotFoundException

Detailed Solution:

In cases when the file specified is not found, then FileNotFoundException is thrown by java run-time system, earlier versions of java used to throw IOException but after Java 2.0 they throw FileNotFoundException.



QUESTION 2:

Which of these values is returned by read() method is end of file (EOF) is encountered?

- a. 0
- b. 1
- c. -1
- d. Null

Correct Answer:

- c. -1

Detailed Solution:

Each time read() is called, it reads a single byte from the file and returns the byte as an integer value. read() returns -1 when the end of the file is encountered.

QUESTION 3:

What will be the output of the following Java program?

```
import java.io.*;

class Chararrayinput {
    public static void main(String[] args) {
        String obj = "abcdef";
        int length = obj.length();
        char c[] = new char[length];
        obj.getChars(0, length, c, 0);
        CharArrayReader input1 = new CharArrayReader(c);
        CharArrayReader input2 = new CharArrayReader(c, 0, 3);
        int i;
        try {
            while ((i = input2.read()) != -1) {
                System.out.print((char) i);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- a. abc** The `getChars` method of the `String` class copies characters from the string `obj` into the character array `c`. This copies characters from index 0 to length - 1 (i.e., the entire string) starting at index 0 of the array `c`.
- b. abcd
- c. abcde
- d. abcdef

Correct Answer:

- a. abc**

Detailed Solution:

A while loop reads characters from `input2` until the end of the stream is reached (i.e., `read()` returns -1). Each character read is cast to a `char` and printed.



QUESTION 4:

What is the purpose of a `ByteArrayOutputStream` in Java?

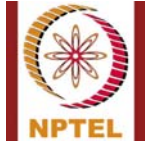
- a. To write binary data to an output stream
- b. To read binary data from an input stream
- c. To convert characters to bytes
- d. To store binary data in memory

Correct Answer:

- d. To store binary data in memory

Detailed Solution:

The `ByteArrayOutputStream` class is used to store binary data in memory. It provides a byte array that can be accessed and manipulated.



QUESTION 5:

`public void write(byte[] b) throws IOException`

Which method is used to read b length bytes from the input stream into an array?

- a. `public void read(int b) throws IOException{}`
- b. `public int read(byte[] b) throws IOException{}`
- c. `public void read(byte[] b) throws IOException{}`
- d. `public int read(int b) throws IOException{}`

Correct Answer:

- b. `public int read(byte[] b) throws IOException{}`

Detailed Solution:

`public int read(byte[] b) throws IOException{}` is used to read b length bytes from the input stream into an array.



QUESTION 6:

Which method is used to create a directory with fileattributes?

- a. `Path.create()`
- b. `Path.createDirectory()`
- c. `Files.createDirectory(path, fileAttributes)`
- d. `Files.create(fileAttributes)`

Correct Answer:

- c. `Files.createDirectory(path, fileAttributes)`

Detailed Solution:

New directory can be created using `Files.createDirectory(path, fileAttribute)`.

QUESTION 7:

What will be the output of the following Java program?

```
public class Calculator {  
    int num = 100;  
  
    public void calc(int num) {  
        this.num = num * 10;  
    }  
  
    public void printNum() {  
        System.out.println(num);  
    }  
  
    public static void main(String[] args) {  
        Calculator obj = new Calculator();  
        obj.calc(2);  
        obj.printNum();  
    }  
}
```

- a. 20
- b. 100
- c. 1000
- d. 2

Correct Answer:

a. 20

Detailed Solution:

Here the class instance variable name(num) is same as calc() method local variable name(num). So for referencing class instance variable from calc() method, this keyword is used. So in statement this.num = num * 10, num represents local variable of the method whose value is 2 and this.num represents class instance variable whose initial value is 100. Now in printNum() method, as it has no local variable whose name is same as class instance variable, so we can directly use num to reference instance variable, although this.num can be used.



QUESTION 8:

Which class is used to write primitive data types to an output stream in Java?

- a. **DataOutputStream**
- b. ObjectOutputStream
- c. OutputStream
- d. PrintWriter

Correct Answer:

- a. **DataOutputStream**

Detailed Solution:

The `DataOutputStream` class is used to write **primitive data** types to an output stream. It provides methods to write **data types like int, double, boolean, etc., to the stream.**

QUESTION 9:

PrintWriter expects a char or a String. Since 106 is an integer, it is converted to its ASCII character equivalent, which is 'j'.

What will be the output of the following code?

```
import java.io.*;

public class W7 {
    public static void main(String[] args) {
        try {

            PrintWriter writer = new PrintWriter(System.out);

            writer.write(9 + 97);

            writer.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

- a. It will give compile-time error
- b. It will give run-time error
- c. j
- d. 106

Correct Answer:

- c. j

Detailed Solution:

The output of this program will be the character 'j' because the Unicode code point for 106 corresponds to 'j'.

QUESTION 10:

What will be the output of the following code.

file.txt contain "This is Programming in Java online course." (without quotes)

```
import java.io.File;

class FileSizeEample {
    public static void main(String[] args) {
        // Specify the file path
        String filePath = "file.txt";

        // Create a File object
        File file = new File(filePath);

        // Get the size of the file
        long fileSize = file.length();

        // Print the size of the file
        System.out.println(fileSize);
    }
}
```

- a. 42
- b. 35
- c. 7
- d. 0

Correct Answer:

- a. 42

Detailed Solution:

The length() method on the File object, which returns the size of the file in bytes.