

Pregunta 1

Given:

```
public class Main {  
    public static void main(String[] args) {  
        try(BufferedReader in = new BufferedReader(new InputStreamReader(System.in))) {  
            System.out.print("Input: ");  
            String input = in.readLine();  
            System.out.println("Echo: " + input);  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

And the command:

```
java Main Helloworld
```

What is the result ?

A. Input:

Echo:

B. Input: Helloworld

Echo: Helloworld

C. Input:

Then block until any input comes from System.in

D. Input:

Echo: Helloworld

E. A NullPointerException is thrown at run time

Pregunta 2

Given:

```
public static void reader(String fileName1) throws Exception{  
    try (var fr = new FileReader(fileName1)); {  
        int charRead = 0;  
        while ((charRead = fr.read()) != -1) {  
            System.out.println("Read char " + charRead);  
        }  
    }  
}
```

What can be done to the above code to make it read Strings instead of chars?

- A. Chain fr to a StringReader and use its readString method.
- B. Use fr.readString instead of fr.read.
- C. Chain fr to a BufferedReader use its readLine method
- D. Chain fr to a DataReader and use its readLine method.

Pregunta 3

Given:

```
public class SerializedMessage implements Serializable {
    String message;
    LocalDateTime createdTime;
    transient LocalDateTime updatedDateTime;;
    SerializedMessage(String message) {
        this.message = message;
        this.createdTime = LocalDateTime.now();
    }
    private void readObject (ObjectInputStream in) {
        try {
            in.defaultReadObject();
            this.updatedDateTime = LocalDateTime.now();
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

When is the readObject method called?

- A. before this object is deserialized
- B. after this object is deserialized**
- C. before this object is serialized
- D. The method is never called.
- E. after this object is serialized

Pregunta 4

Given:

Given that the file test.txt contains :

12345678

What will the following code print when compiled and run?

```
public static void main(String[] args) throws Exception{  
    try(var fis = new FileInputStream("c:\\temp\\test.txt");  
        var isr = new InputStreamReader(fis)){  
        while(isr.ready()){  
            isr.skip(1);  
            int i = isr.read();  
            char c = (char) i;  
            System.out.print(c);  
        }  
    }  
}
```

When run and all three files exist, what is the state of each reader on Line 1?

- A. It will not compile.
- B. It will throw an exception when run.
- C. It will print just 2.
- D. It will run without any exception but will not print anything.
- E. It will print 2468**

Pregunta 5

What is the output of the following program? Assume the file paths referenced in the class exist and are able to be written to and read from.

```
import java.io.*;

public class Vegetable implements Serializable {

    private Integer size = 1;

    private transient String name = "Red";

    { size = 3; name = "Purple"; }

    public Vegetable() { this.size = 2; name = "Green"; }

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

        try (var o = new ObjectOutputStream(
            new FileOutputStream("healthy.txt"))) {

            final var v = new Vegetable();

            v.size = 4;

            o.writeObject(v);

        }

        try (var o = new ObjectInputStream(
            new FileInputStream("healthy.txt"))) {

            var v = (Vegetable) o.readObject();

            System.out.print(v.size + "," + v.name);

        } } }
```

- A. 1,Red
- B. 2,Green
- C. 2,null
- D. 3,Purple
- E. 4,null**
- F. null,null