1. Mark all options that are considered an Input Streams in Java:

|  |  |  |
| --- | --- | --- |
| ( ) System.out | ( ) System.in | ( ) String |
| ( ) File | ( ) FileOutput | ( ) FileInput |

1. Write the exact output of the program below, if we run it.

public static void example4(){  
 String line = "This is a message\nComposed by\nThree lines";  
 Scanner scanner = new Scanner(line);  
 int tokenCounter = 0;  
 while(scanner.hasNext()) {  
 tokenCounter++;  
 System.*out*.println(scanner.next());  
 }  
 System.*out*.println(tokenCounter);  
}

1. Write the exact output of the program below, if we run it.

public static void example5(){  
 String line = "This is a message\nComposed by\nThree lines";  
 Scanner scanner = new Scanner(line);  
 int tokenCounter = 0;  
 while(scanner.hasNext()) {  
 tokenCounter++;  
 System.*out*.println(scanner.nextLine());  
 }  
 System.*out*.println(tokenCounter);  
}

1. Assuming that you have a class Person with the following constructor and a txt file with the following format, analyze the code below and explain if it is correct or incorrect.

|  |  |
| --- | --- |
| public Person(String name, int years){  setName(name);  setYearsOld(years); } |  |

public static ArrayList<Person> readFilePerson(String fileName){  
 ArrayList<Person> list = new ArrayList<>();  
 try {  
 Scanner fileIn = new Scanner(new File(fileName));  
 String name = "";  
 int years = 0;  
 while(fileIn.hasNext()) {   
 String line = fileIn.nextLine();  
 Scanner sLine = new Scanner(line);  
 sLine.useDelimiter(",");  
 String nameS = sLine.next();  
 String yearsS = sLine.next();  
 name = nameS.substring(0, nameS.indexOf(":"));  
 years = Integer.*parseInt*(yearsS.substring(yearsS.indexOf(":")+1));  
 Person p = new Person(name, years);  
 list.add(p);  
 }  
 }catch(IOException ex) {  
 System.*err*.print("Error reading file!");  
 ex.printStackTrace();  
 }  
 return list;  
}