

Forum:[Hack Forums](#)

Thread: [Variable converting to ASCII for no reason](#)

CubicMuffin - November 5, 2015, 7:07 pm

Hello all, it's my first post in this section but I've been using Java for a year or so now. I'm creating this program for working out Hamming Code parity bits, and it works perfectly with manual input but I'm trying to get it just read bitstreams from a text file and do it that way. Which works... Almost.

```
Code for (int x = 0; x < txtrfInput[0].length(); x++){
    bitInput[x] = txtrfInput[0].charAt(x);
    bitLength = txtrfInput[0].length();
    System.out.println("bl: " + bitInput[x]);
    System.out.println("tl: " + txtrfInput[0].charAt(x));
}
```

Now, it's only reading in 1s and 0s because it's binary, but on the first System.out.println line (for testing) which displays the value of bitInput, it comes out with 48 or 49, which is the ASCII code for 1 or 0. However, when looking at the second print line for txtrfInput (just a temp name, ignore bad formatting) it displays 0 or 1.

Anyone know why it does this conversion?
Thanks!

EDIT:

Here's the relevant code, if that helps:

```
Code //Automatic Input
//Reading in text input
String file_name = "E:/University/Term 1/CaCT/Week
5/CorrectParityChecker/src/ParityBitter/bitStreams.txt";
String[] txtrfInput = new String[10];
try {
    ReadFile file = new ReadFile(file_name);
    String[] txtInput = file.OpenFile();
    txtrfInput = txtInput;
}
catch (IOException e){
    System.out.println("Error. The file doesn't exist :( or I've messed up the code.");
}

for (int x = 0; x < txtrfInput[0].length(); x++){
    bitInput[x] = txtrfInput[0].charAt(x);
    bitLength = txtrfInput[0].length();
    System.out.println("bl: " + bitInput[x]);
    System.out.println("tl: " + txtrfInput[0].charAt(x));
}
```

Cathal - November 6, 2015, 1:45 am

Nothing jumps out at me immediately.. Can you post the source of ReadFile? By the way I strongly recommend you read this <http://google-styleguide.googlecode.com/svn/trunk/javaguide.html>

CubicMuffin - November 6, 2015, 1:49 am

(Nov 6, 2015 02:45 AM)Cathal Wrote: [\[CITE\]](#) Nothing jumps out at me immediately.. Can you post the source of ReadFile? By the way I strongly recommend you read this <http://google-styleguide.googlecode.com/svn/trunk/javaguide.html>
Don't think it has anything to do with that but alright:

```
Code public class ReadFile {  
  
    private String path;  
  
    public ReadFile(String file_path){  
        path = file_path;  
    }  
  
    public String[] OpenFile() throws IOException{  
  
        FileReader fr = new FileReader(path);  
        BufferedReader textReader = new BufferedReader(fr);  
  
        String[] textData = new String[10];  
  
        for(int i = 0; i < textData.length; i++){  
            textData[i] = textReader.readLine();  
        }  
  
        textReader.close( );  
        return textData;  
    }  
}
```

Also any specific reason why I should read it (the guide)? Just so I know what I'm looking for if anything.

Cathal - November 6, 2015, 1:56 am

(Nov 6, 2015 02:49 AM)CubicMuffin Wrote: [\[CITE\]](#) Don't think it has anything to do with that but alright:

```
Code public class ReadFile {  
  
    private String path;  
  
    public ReadFile(String file_path){  
        path = file_path;  
    }  
  
    public String[] OpenFile() throws IOException{  
  
        FileReader fr = new FileReader(path);  
        BufferedReader textReader = new BufferedReader(fr);  
  
        String[] textData = new String[10];  
  
        for(int i = 0; i < textData.length; i++){  
            textData[i] = textReader.readLine();  
        }  
    }  
}
```

```

    }

    textReader.close();
    return textData;
}
}

```

Also any specific reason why I should read it (the guide)? Just so I know what I'm looking for if anything.

As far as I know, the `BufferedReader` class will automatically attempt to convert raw bytes to characters. The class is designed to parse char streams, but you don't want that. If you want to read the file as a string just use `Scanner`.

I asked you to look at Google's style guide because your code formatting is extremely inconsistent!

CubicMuffin - November 6, 2015, 1:59 am

(Nov 6, 2015 02:56 AM)Cathal Wrote: [\[CITE\]](#) As far as I know, the `BufferedReader` class will automatically attempt to convert raw bytes to characters. The class is designed to read text, not binary. If you want to read the file as a string just use `Scanner`.

I asked you to look at Google's style guide because your code formatting is extremely inconsistent! It's text that I want. I read the binary in as a `String`, it just so happens the string is either going to be a "0" or a "1". Also it doesn't explain why `txtCrLfInput` displays the correct string but `bitInput` which is literally told to be assigned the value of `txtCrLfInput` converts it to ASCII! Very confusing.

And I didn't think it was very inconsistent. If it's indentation you are talking about then copying and pasting code into here makes it mess up so I have to guess. Other than that, I thought it looked fine. (Also most of the first bit of code I sent you is still in testing so names etc aren't really going to be like that on the finished thing)

Cathal - November 6, 2015, 2:14 am

(Nov 6, 2015 02:59 AM)CubicMuffin Wrote: [\[CITE\]](#) It's text that I want. I read the binary in as a `String`, it just so happens the string is either going to be a "0" or a "1". Also it doesn't explain why `txtCrLfInput` displays the correct string but `bitInput` which is literally told to be assigned the value of `txtCrLfInput` converts it to ASCII! Very confusing.

And I didn't think it was very inconsistent. If it's indentation you are talking about then copying and pasting code into here makes it mess up so I have to guess. Other than that, I thought it looked fine. (Also most of the first bit of code I sent you is still in testing so names etc aren't really going to be like that on the finished thing)

Yeah i understand ya now, to be honest I don't know how to help you, sorry man!

As for the formatting, it's not the indentation I was looking at. In one case you name a variable using snake_case (`file_name`), but you use camelCase just a bit later (`txtInput`).

In some places you leave a space before an opening curly bracket, but other times you don't.
Code try {

...

catch (IOException e){

Lastly, it's common practice to use `i`, `j` and `k` for iterators or indices. Never `x`!

Obviously none of these are game-breaking problems but you should fix bad habits early.

CubicMuffin - November 6, 2015, 2:21 am

(Nov 6, 2015 03:14 AM)Cathal Wrote: [\[CITE\]](#) Yeah i understand ya now, to be honest I don't know how to help you, sorry man!

As for the formatting, it's not the indentation I was looking at. In one case you name a variable using snake_case (file_name), but you use camelCase just a bit later (txtInput).

In some places you leave a space before an opening curly bracket, but other times you don't.
Code try {

...
catch (IOException e){

Lastly, it's common practice to use i, j and k for iterators or indices. Never x!

Obviously none of these are game-breaking problems but you should fix bad habits early.

The snake case is because I was just copying a tutorial for reading in text (I was a bit rusty on that) so I haven't got round to changing the names. I didn't realise there was a space there between "try" and "{", I always try and put it directly afterwards so thanks for pointing that out!

And yeah I use i/j/k in literally every other iterator except that one :P

Again, it's just for testing and so wasn't really "proper code". Glad it wasn't anything serious though! Thanks for trying to help anyway, I'll probably just try something else if I can't get it.

clrlslwi - November 6, 2015, 3:08 am

The only thing I can figure is that bitInput is a numerical array, causing the character to be casted to a number. You didn't post all of your code though, so I'm not sure.

CubicMuffin - November 6, 2015, 10:04 am

(Nov 6, 2015 04:08 AM)Demmonic Wrote: [\[CITE\]](#) The only thing I can figure is that bitInput is a numerical array, causing the character to be casted to a number. You didn't post all of your code though, so I'm not sure.

[Heres the code](#) for the main class.

Just a heads up, it's filled with random-ass comments because that's how I think :P You shouldn't really need to look at anything below the first input section.

clrlslwi - November 6, 2015, 10:07 am

(Nov 6, 2015 11:04 AM)CubicMuffin Wrote: [\[CITE\]](#) [Heres the code](#) for the main class.

Just a heads up, it's filled with random-ass comments because that's how I think :P You shouldn't really need to look at anything below the first input section.

Code int[] bitInput = new int[256];

It prints the ASCII value because you've stored it in an int array, causing the JVM to treat it as an int. You can cast it back to print the character itself. Or just make it an array of characters.

Code System.out.println("bl: " + (char)bitInput[x]);

CubicMuffin - November 6, 2015, 10:16 am

(Nov 6, 2015 11:07 AM)Demmonic Wrote: [\[CITE\]](#) Code int[] bitInput = new int[256];

It prints the ASCII value because you've stored it in an int array, causing the JVM to treat it as an int. You can cast it back to print the character itself. Or just make it an array of characters.

Code System.out.println("bl: " + (char)bitInput[x]);

Ah, okay, thanks for your help!