

Nihar Gupte  
1001556441  
Faults Detected and Corrections Report

- “Line Number” column represents the location of the fault in the original “printtokens.java” file used for the project.
- “Fault Detected” column represents the actual code wherein the fault lies.
- “Correction to fix” column represents the changes to code that have to be made in the detected fault to fix it.
- “Notes” column represents additional comments such as the reason why the line had a fault, or the reason behind changing the code, or a missing statement in the code.
- Missing code blocks are given at the end of the document.

---

#	Line Number	Fault Detected	Correction to fix	Notes
1	30	if (fname == null)	if (fname.equals(""))	fname has two possible values – "" or filename. It is never passed on as null
2	42	return null;	return br;	returning the value of BufferedReader br, instead of null everytime
3	86	if(fname.equals(null))	if(fname==null)	Incorrect syntax in java to check for null condition
4	87	br=open_character_stream(null);	br=open_character_stream("");	According to comments in main(), "" should be passed when no filename is specified, not null
5	124	if(ch == "")id=2;	if(ch == "")id=1;	According to conditions in is_token_end, id for string token is 1, and id for comment token is 2
6	125	if(ch ==59)id=1;	if(ch ==59)id=2;	
7	249	Missing code	*Refer to missing block #1 at the end of this document	if conditions to check for string constant and comment are missing
8	289	if (str.length() > 2 && ...)	if (str.length() == 2 && ...)	Length of character constants is 2, not greater than 2
9	306	while ( i <= str.length() && ...)	while ( i < str.length() && ...)	Preventing StringIndexOutOfBoundsException in Java

10	308	if(Character.isDigit(str.charAt(i+1)))	if(Character.isDigit(str.charAt(i)))	Preventing IndexOutOfBoundsException in Java
11	329	while (... && str.charAt(0)!='\0')	while (... && str.charAt(i)!='\0')	null terminator has to be checked at the i'th index, not the starting index
12	335	return true;	return false;	Return false if the string constant does not end in quotes ""
13	359	return false;	return true;	Return true if identifier condition followed
14	362	return true;	return false;	Return false if token does not start with a letter
15	383	Missing code	*Refer to missing block #2 at the end of this document	if condition to check for special character comma is missing
16	383	if (str.equals("{"))	if (str.equals("("))	“{“ used in place of “(“
17	417	Missing code	*Refer to missing block #3 at the end of this document	if condition to check for special character “/” is missing
18	456	Missing code	*Refer to missing block #4 at the end of this document	if condition missing to check for special character single quote '
19	462	fname = new String();	//fname = new String();	Line commented so that fname gets passed on to open_token_stream as null, and hence changes later on to "" – the way the control flow is set up
20	464	fname = args[1];	fname = args[0];	Changed to 0 <sup>th</sup> index, as the filename is saved in 0 <sup>th</sup> index
21	467	System.exit(0);	return;	Using a pseudo return instead of system.exit() so that JUnit does not terminate
22	477	System.exit(0);	return;	

### **List of missing blocks of code:**

#### **1. Missing block #1**

```

if(type == str_constant) {
    System.out.print("string," + tok + ".\n");
}
if(type == comment) {
    System.out.print("comment,"+tok+".\n");
}

```

```
}
```

## 2. **Missing block #2**

```
if(str.equals(",")) {  
    System.out.print("comma.\n");  
    return;  
}
```

## 3. **Missing block #3**

```
if (str.equals("/")){  
    System.out.print("slash.\n");  
    return;  
}
```

## 4. **Missing block #4**

```
if(c == "\\") return true;
```