

---

## Common Errors

There are certain common errors and exceptions which beginners come across and find them very annoying. Here we will discuss these and give a little explanation of what's going wrong and how you can avoid that. Given below is a list of common mistakes along with their explanation.

- `NoClassDefFoundError`
- `NoSuchMethodError`
- Cannot resolve symbol
- Local variables must be initialized
- `ArrayIndexOutOfBoundsException` & `NumberFormatException` exception
- `NullPointerException`
- Unreachable Code

---

## NoClassDefFoundError

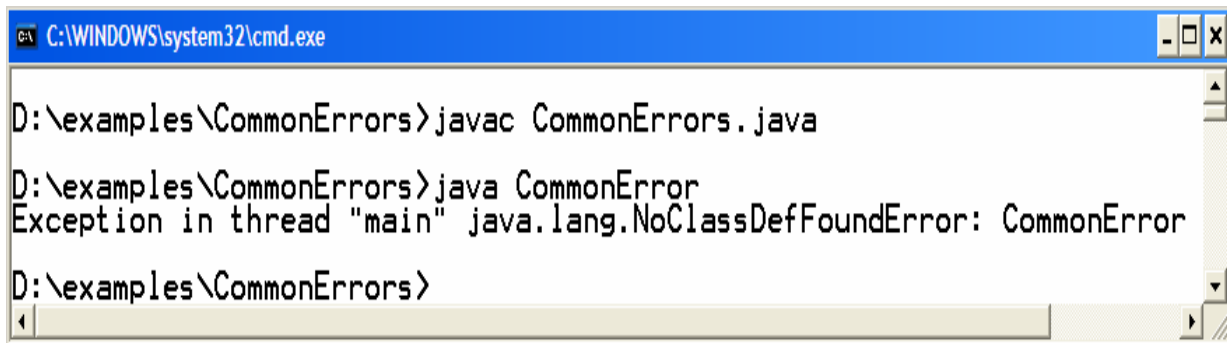
### Problem

This error occurs when the class you are trying to run a class which does not exists there. It could be due to wrong name that you are specifying or may be you are in the wrong directory. (e.g. you have saved the file in ex1 directory and trying to run it from ex2 directory)

### Example

In the example below the name of the class is “CommonErrors”. The user has compiled the class correctly but running it using a wrong name “CommonError”.

```
public class CommonErrors{  
    public static void main (String args[]) {  
        System.out.println("Hello World");  
    }  
}
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The user is in the directory "D:\examples\CommonErrors". They first compile the class using "javac CommonErrors.java". Then, they attempt to run the class using "java CommonError". This results in a "NoClassDefFoundError: CommonError" exception, indicating that the class name specified in the command does not match the actual class name "CommonErrors".

```
C:\WINDOWS\system32\cmd.exe  
D:\examples\CommonErrors>javac CommonErrors.java  
D:\examples\CommonErrors>java CommonError  
Exception in thread "main" java.lang.NoClassDefFoundError: CommonError  
D:\examples\CommonErrors>
```

---

## NoSuchMethodError:main

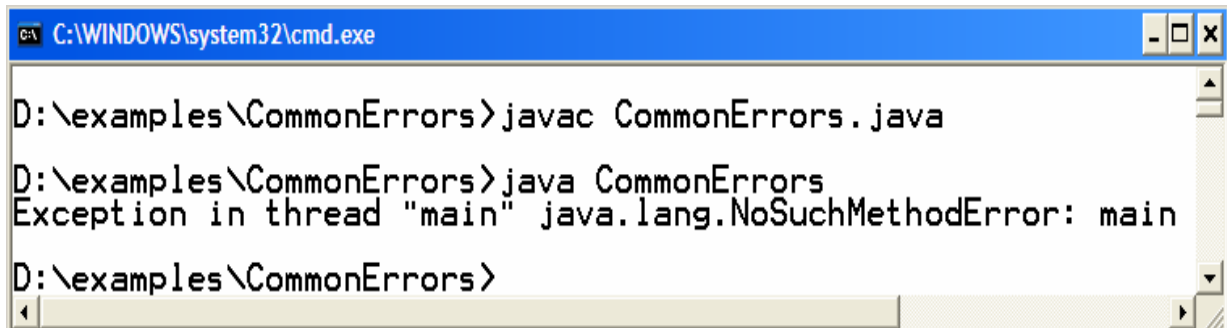
### Problem

You are trying to run a class which does not have the main method. main() is where your code starts to run and if a class does not have the main method and you try to run it you will get this error.

### Example

The following code does not contain the main method and when the user is trying to run that code, he/she will get the error that we have specified above

```
public class CommonErrors{  
  
    public static void display () {  
        System.out.println("hello world");  
    }  
  
}
```



```
C:\WINDOWS\system32\cmd.exe  
  
D:\examples\CommonErrors>javac CommonErrors.java  
D:\examples\CommonErrors>java CommonErrors  
Exception in thread "main" java.lang.NoSuchMethodError: main  
D:\examples\CommonErrors>
```

---

## Cannot resolve Symbol

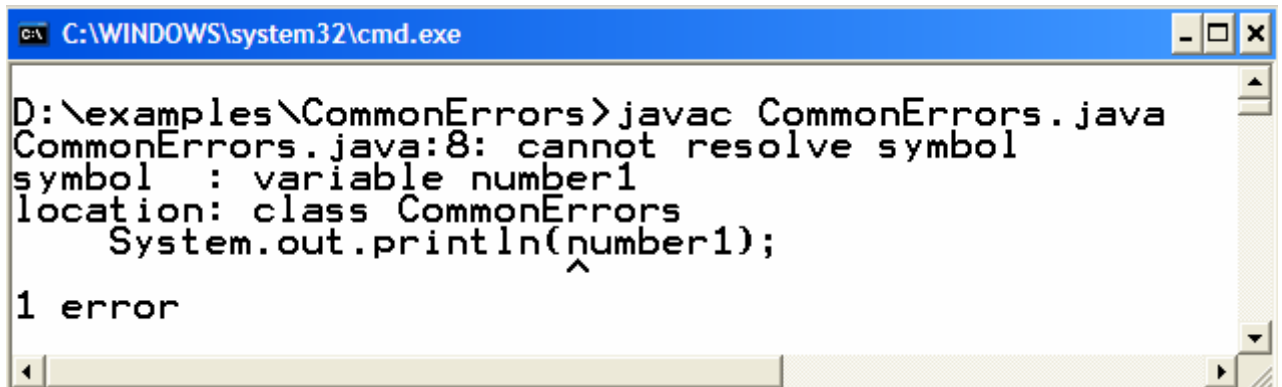
### Problem

You are trying to use a variable or method does not exist. It can be due to incorrect name, wrong number or order of arguments etc.

### Example

In the example below we have declared variable with the name “number” but we are trying to use the variable with the name “number1”

```
public class CommonErrors{  
  
    public static void main (String args[]) {  
  
        int number;  
  
        System.out.println(number1);  
    }  
}
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The command prompt displays the following text:

```
D:\examples\CommonErrors>javac CommonErrors.java  
CommonErrors.java:8: cannot resolve symbol  
symbol  : variable number1  
location: class CommonErrors  
    System.out.println(number1);  
                        ^  
1 error
```

---

## Variable might not have been initialized

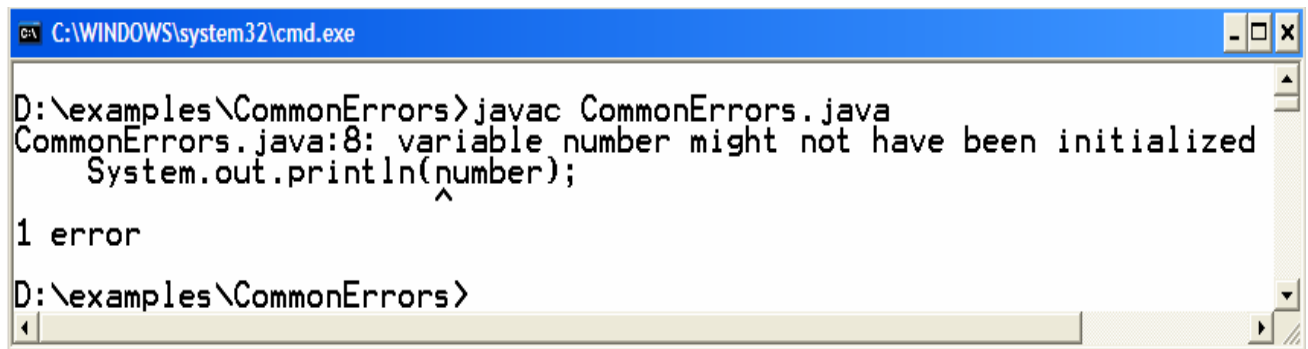
### Problem

In java variables declared at class level, get initialized automatically but the variables inside a method does not get initialized automatically. Now if we try to use any such variable as a right value (if we try to assign its value to some other variable) or in a method, we will get this error.

### Example

In the example given below variable number is declared inside method main and is not initialized and you can see that when we are trying to print its value we are getting this error.

```
public class CommonErrors{  
  
    public static void main (String args[]) {  
  
        int number ;  
  
        System.out.println(number);  
    }  
}
```



The screenshot shows a Windows command prompt window with the title bar "C:\WINDOWS\system32\cmd.exe". The command prompt shows the following text:

```
D:\examples\CommonErrors>javac CommonErrors.java  
CommonErrors.java:8: variable number might not have been initialized  
    System.out.println(number);  
                        ^  
1 error  
D:\examples\CommonErrors>
```

---

## ArrayIndexOutOfBoundsException

### Problem:

Arrays are bound checked in java. It means that you can not access an index of an array which does not exist. If you try to do that this exception will be thrown

### Example

In the example below we are trying to access the first index of the args[] array but while running the program we have not provided that index in the first run and hence it caused an exception but in the second run when we have provided a value exception did not occur

```
public class CommonErrors{  
    public static void main (String args[]) {  
        System.out.println(args[0]);  
    }  
}
```



```
C:\WINDOWS\system32\cmd.exe  
D:\examples\CommonErrors>javac CommonErrors.java  
D:\examples\CommonErrors>java CommonErrors  
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0  
    at CommonErrors.main(CommonErrors.java:4)  
D:\examples\CommonErrors>java CommonErrors hello  
hello  
D:\examples\CommonErrors>_
```

---

## NumberFormatExceptions

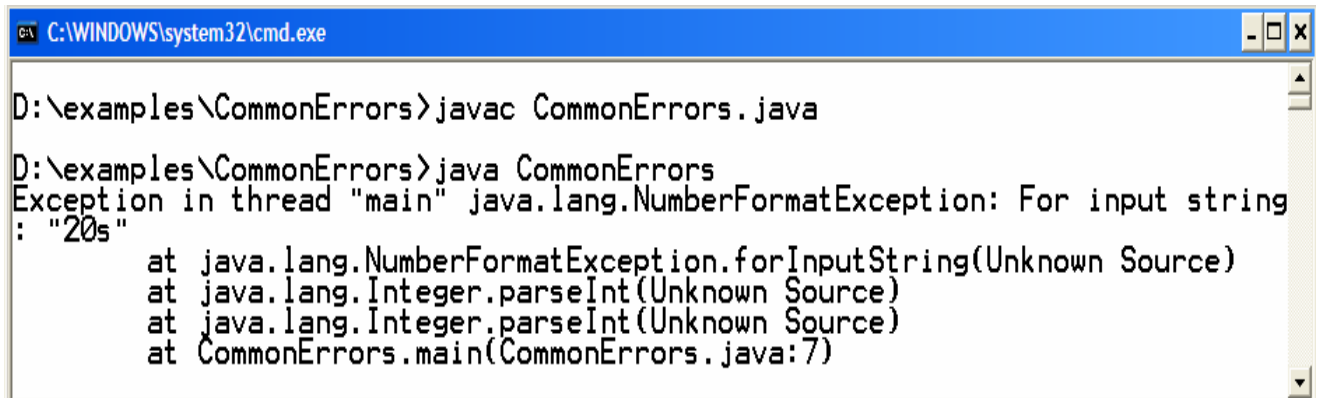
### Problem

You are trying to convert a string or an object into a number, which is not actually a number, `NumberFormatException` is thrown

### Example

In the example below, we are trying to convert a string “20s” into a number. As you can figure out that this is not a number and hence exception will occur.

```
public class CommonErrors{  
  
    public static void main (String args[]) {  
  
        String s = “20s”;  
  
        int number = Integer.parseInt(s);  
    }  
}
```



```
C:\WINDOWS\system32\cmd.exe  
D:\examples\CommonErrors>javac CommonErrors.java  
D:\examples\CommonErrors>java CommonErrors  
Exception in thread "main" java.lang.NumberFormatException: For input string  
: "20s"  
    at java.lang.NumberFormatException.forInputString(Unknown Source)  
    at java.lang.Integer.parseInt(Unknown Source)  
    at java.lang.Integer.parseInt(Unknown Source)  
    at CommonErrors.main(CommonErrors.java:7)
```

---

## NullPointerException

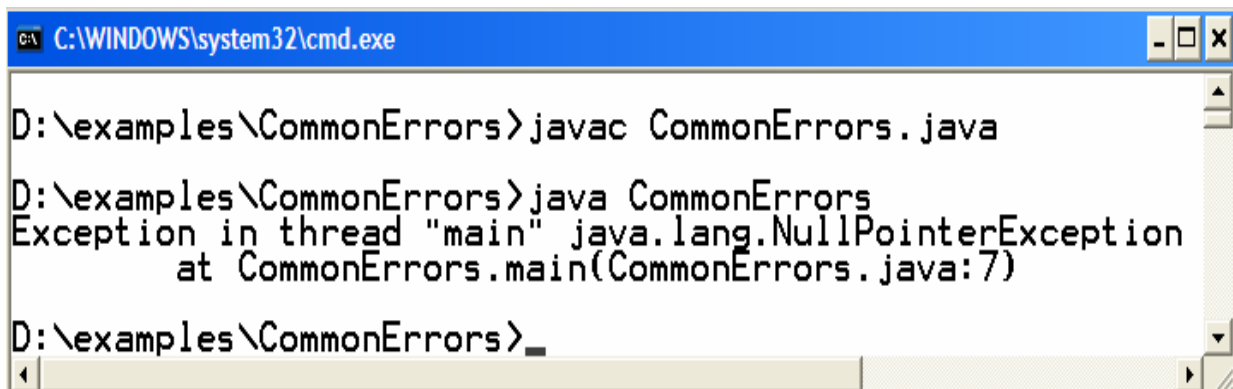
### Problem:

You are trying to use a reference which is not pointing to any object. This is a very common problem with the beginners and usually occurs due to the failure of calling constructor to initialize an object.

### Example

In the example below a variable 's' of type 'Student' is declared, but its constructor is not called and hence 's' is not pointing to any object. Now if we call a method on 's' then 'NullPointerException' will be thrown because is pointing to null.

```
public class CommonErrors{  
  
    public static void main (String args[ ]) {  
  
        Student s = null;  
  
        s.print();  
    }  
}
```



```
C:\WINDOWS\system32\cmd.exe  
  
D:\examples\CommonErrors>javac CommonErrors.java  
  
D:\examples\CommonErrors>java CommonErrors  
Exception in thread "main" java.lang.NullPointerException  
    at CommonErrors.main(CommonErrors.java:7)  
  
D:\examples\CommonErrors>_
```