

**Concordia University**

SOEN 6011 - SOFTWARE ENGINEERING PROCESS

---

**ETERNITY : FUNCTIONS**  
**Function 6 : B(x,y)**

---

Problem Solution 4

**Juhi Birju Patel**

Student ID : 40190446

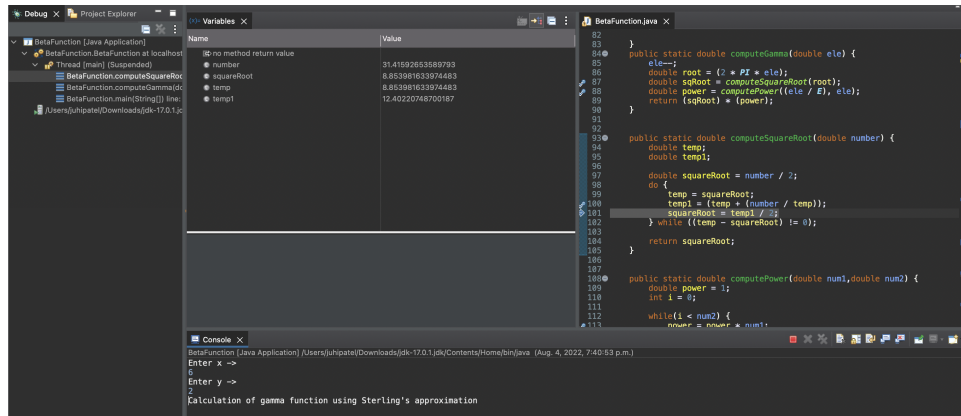
<https://github.com/JuhiCodes/SOEN-6011-Course-Project>

# Contents

1	Debugger - Eclipse . . . . .	2
2	Quality Attributes . . . . .	3
3	Source Code review Checkstyle . . . . .	3

# 1 Debugger - Eclipse

Debugging for the code is performed using eclipse debugger. Several break-points are investigated.



The Eclipse Java IDE has many debugging tools and views present in the Debug Perspective to help the developer debug the code effectively and efficiently. A developer can run an application in Debug mode and can iterate over each line of code in a program. In addition to this, Eclipse has a Debug Perspective which is a set of views grouped together that can be helpful to inspect code and make the debugging process very effective.

## Advantages

- Eclipse allows you to many event based breakpoints.
- It provides with filtering options like step into, step over etc.
- The values of the variables under inspection can be changed while debugging.
- Developer can view a logical structure of their code.

## Disadvantages

- Debugging when we have multithreaded programs is difficult.
- When the program is huge, debugging is not effective with the debugger.

## 2 Quality Attributes

Below mentioned are the steps taken to ensure the quality of the source code.

### Correctness

To check for the correctness of the code, all the input values are validated. The results are compared to the results received from a scientific calculator.

### Robust

Error handling is used to validate proper inputs. In addition to this, unit testing is performed on the code.

### Time Efficient

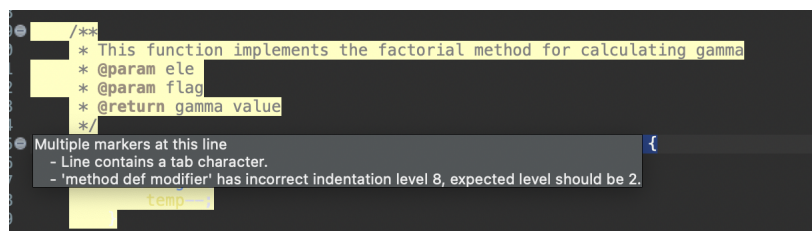
The big O complexity of the code is quite low which makes the code run faster.

### Usable

The interface is easy to use. Proper messages are displayed on events of taking user inputs and display of error messages. The results are also formatted properly which makes it easy for the user to interpret them.

## 3 Source Code review Checkstyle

Checkstyle is a development tool that aims at helping developers to write Java code that follows a coding standard. It automates the process of checking Java code.



The following issues were observed:

- Most of the error were indentation errors, which were fixed.
- JavaDoc comments were missing which were added.
- Else statements were to be added with every if statement.
- Variable naming did not follow standard naming practice, which was corrected.

# Bibliography

- [1] Eclipse Debugger,  
[https://www.eclipse.org/community/eclipse\\_newsletter/2017/june/article1.php](https://www.eclipse.org/community/eclipse_newsletter/2017/june/article1.php)
- [2] Checkstyle,  
<https://checkstyle.sourceforge.io>