

SOEN6011 Project: Problem-4

Function: $B(x,y)$

40197897-Pal Patel

4.1 Programming Style

Coding Style is the set of guidelines to be followed while implementing the source code for a software program[1].

4.2 Error handling

Exception handling is the process of responding to unwanted or unexpected events when a computer program runs[3]. When an error occurs, the situation needs to be handled by appropriate error exception strategy. In some cases, the system might even crash if the exceptions are not handled properly.

In the code attached with the submission, try catch blocks have been including right from the start where the inputs to the beta function are being taken. If the user does not enter numbers and enters text, for instance, he would have an Input Mismatch exception. Similarly, the exception where any one or both of the inputs are negative should be caught in the catch block there after.

```
*/
public static void main(String[] args) {
    // TODO Auto-generated method stub
    try {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter input x");
        String stringx= in.nextLine();
        double x = Double.parseDouble(stringx);
        System.out.println("Enter input y");
        String stringy= in.nextLine();
        double y = Double.parseDouble(stringy);
        double beta = (gamma(x) * gamma(y))/gamma(x+y);
        System.out.println("The result of the beta function with inputs x= " + x + " and y= " + y);
    }
    catch(InputMismatchException e){
        System.out.println("Invalid input. Please enter a positive real number");
    }
    catch(Exception e){
        System.out.println(e);
    }
}
```

Figure 1: Exception Handling as per implementation

4.3 Debugger

A debugger or debugging tool is a computer program used to test and debug other programs (the "target" program). [2]. Debugging allows the programmer to track the operations of the program and to monitor the changes in the resources used. Debugging is a very useful tool to fix bugs and logical errors/. For the project, the debugger in built in the Eclipse IDE has been used.

Benefits of using the Eclipse Debugger

- The eclipse debugging tool is easy to use.
- The logs provided by the debugger are valid and sufficient enough of bug fixing.
- The tool is easily available to all programmers for use.

Limitations of the Eclipse Debugger

- Complex to use for large applications and codes.

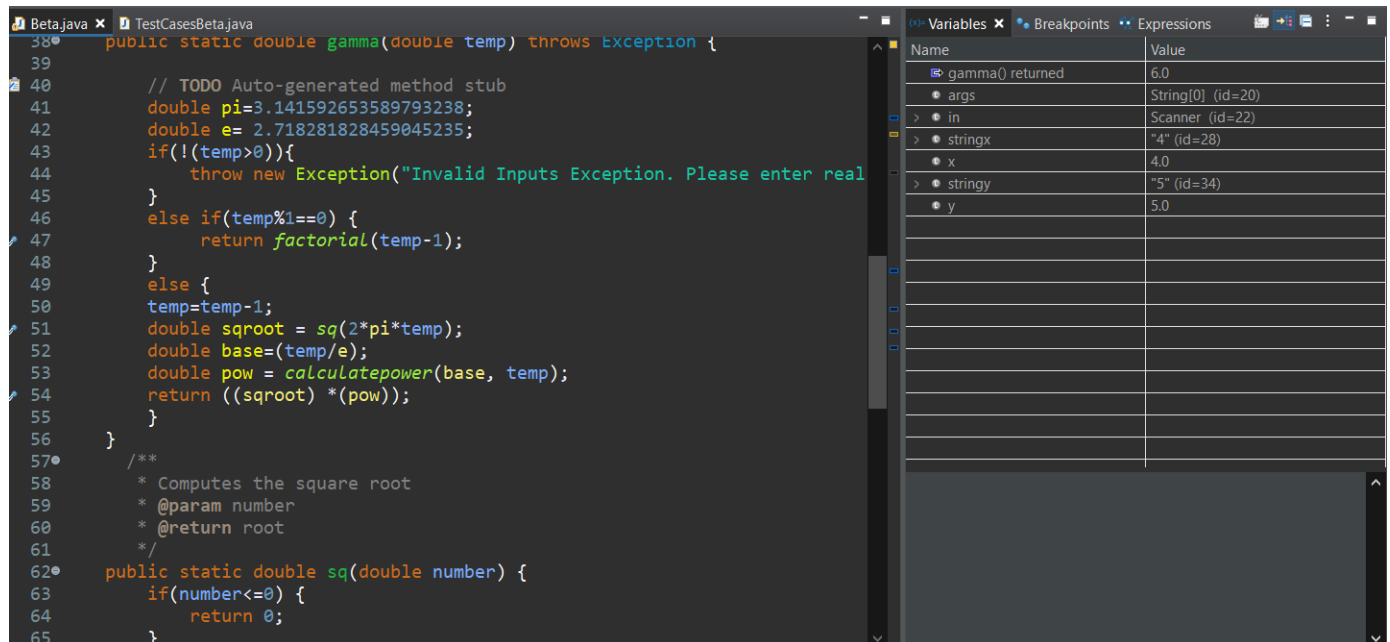


Figure 2: Eclipse Debugger

4.4 Quality Attributes

1. Portable

Java has the feature of WORA- Write Once Read Anywhere, meaning that a bytecode obtained after compilation can be run on any platform easily.

2. Space-efficient

While using the factorial method to find the calculation for the beta function, tail recursion has been used for calculating the factorial. This would ensure that the stack does not overflow.

3. Correct

All the test cases were passed for the functions building the code and hence the code can be claimed correct.

4. Robust

The entered data is validated and exceptions are checked for and thus the code is robust.

5. Maintainable

The entire code has been broken down into appropriate functions and calls have been made as necessary. This makes it easy for any programmer to change it without much hassle.

6. Readable

The code is simply written and put. It is readable enough for any programmer to understand and modify accordingly.

Appropriate comment lines are given for clarification purposes.

7. Time Efficient

The testing of the code takes only 1 millisecond to run 4 test cases of different kinds.

4.5 CheckStyle

Checkstyle is a coding tool that keeps the programmers adhere to the coding standards followed globally. Checkstyle can check many aspects of your source code. It can find class design problems, method design problems. It also has the ability to check code layout and formatting issues [4].

The checkstyle tool on the Eclipse IDE has been used on my code. **Benefits of CheckStyle:**

- Compliance in the program remains in check.
- Ability of creating your own rules. Eclipse defines a large set of styles, but checkstyle has more, and you can add your own custom rules [5]

Limitations of CheckStyle:

- Checkstyle is a single file static analysis tool [4]

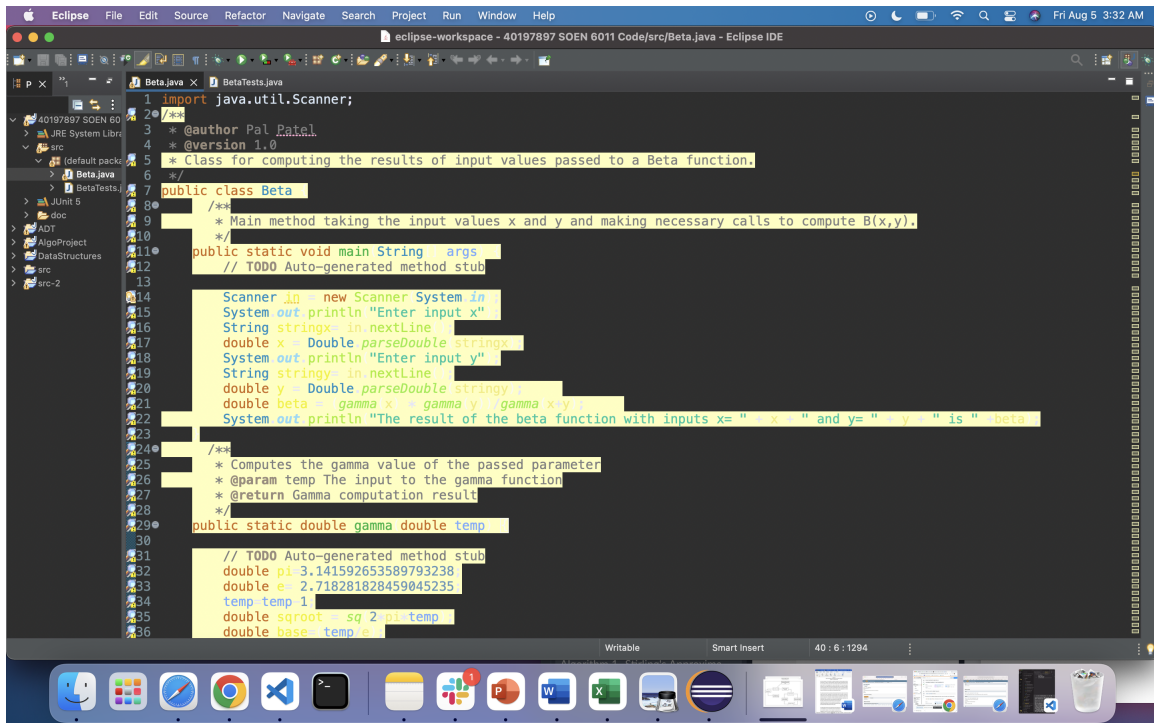


Figure 3: CheckStyle Errors 1

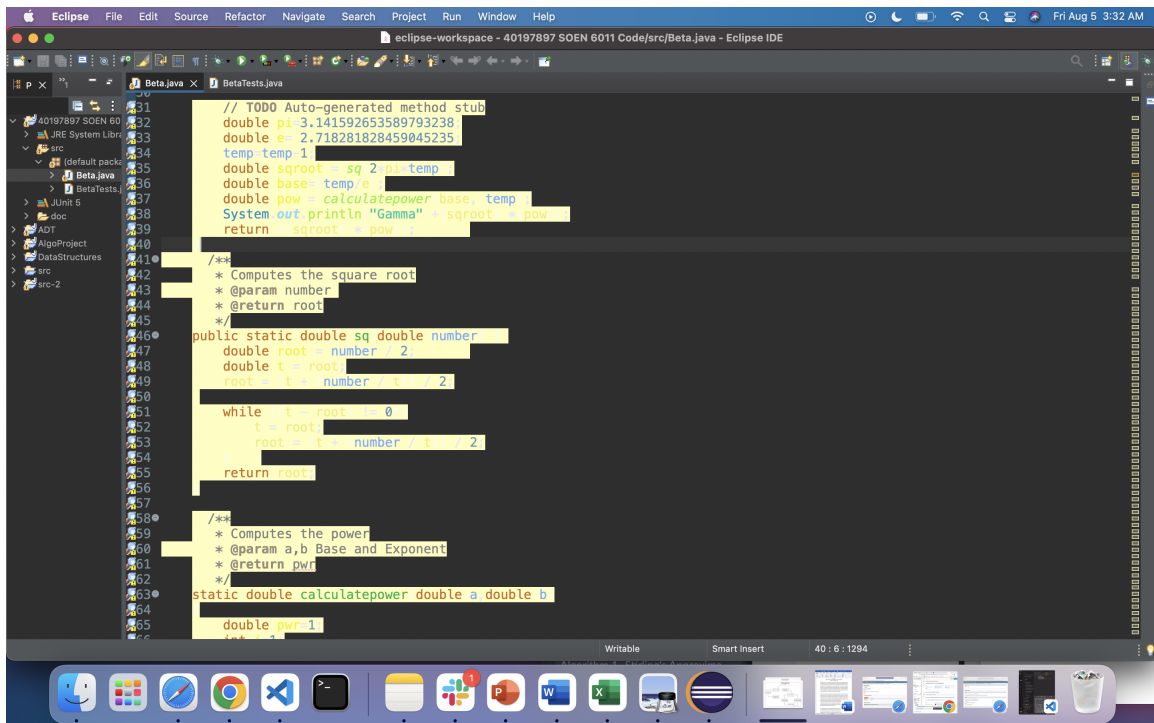


Figure 4: CheckStyle Errors 2

Bibliography

[1] https://en.wikipedia.org/wiki/Programming_style

[2] <https://en.wikipedia.org/wiki/Debugger>

[3] <https://www.techtarget.com/searchsoftwarequality/definition/error-handling>

[4] <https://checkstyle.sourceforge.io/>

[5] <https://stackoverflow.com/questions/13644624/advantage-of-using-checkstyle-rather-than-using-eclipse>