Software Engineering Process - SOEN 6011
Project Report

Saswati Chowdhury
ID : 40184906

5th August 2022

Problem-1

1.1 Introduction

The tangent(tan) function, one of the primary six trigonometric functions, is typically expressed as $tan(x)$**[1]**.
The function as a ratio of the sine function to the cosine function, which can be obtained using a unit circle, is one way to define the tangent function**[1]**. The equation is as follows:

$tan(\theta) = \frac{sin(\theta)}{cos(\theta)}$

It is the also the ratio of the angle's opposite and adjacent sides when a right-angled triangle is being considered**[2]**. This can be referred to as:

$tan(\theta) = \frac{OppositeSide}{AdjacentSide}$

Characteristics[1][2][5][6]

- Since tangent has the form $f(-x) = -f(x)$, it can be regarded as an odd function **[3]**.
- The tangent function is undefinable when $x = \pi/2 + n\pi$ for which, cos(x) = 0 (where, n is integer)
- An intersecting line appears in both x and y axes at (0, 0)
- tan(x) is symmetric in nature.

- Period : π
- Below is a graph of the tangent function tan(x) [**4**]:

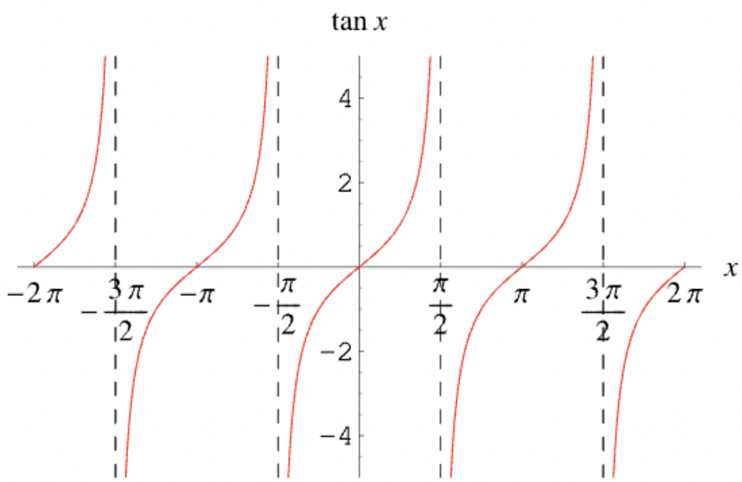

Figure 1: Graph of tangent function - $tan(x)$

Domain and Co-Domain[5]

There are no values where cos x equals 0 in the $tan(x)$ domain. Hence, since $cos(x)$ is 0 at odd multiples of $\pi/2$, the domain and Co-Domain of tangent function can be expressed as follows: .

1. Domain : $\{x \mid x \neq \frac{\pi}{2} + k\pi, k = ..., -1, 0, 1, ...\}$
2. Co-Domain : R

1.2 Context of Use Model for the Tangent Calculator

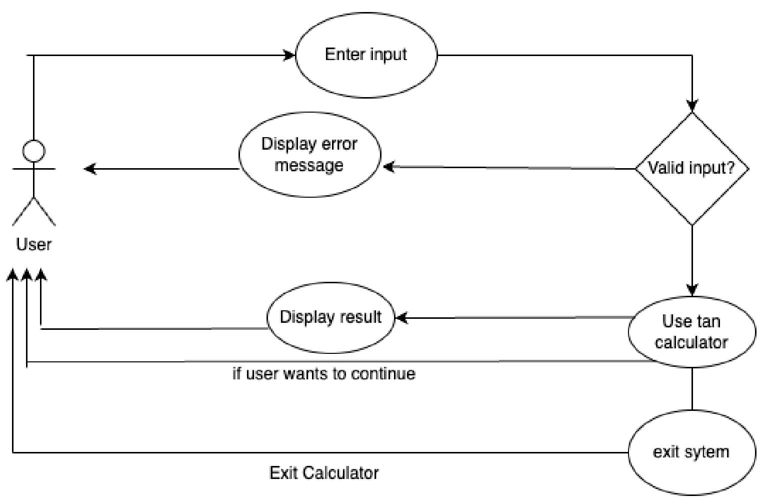

Figure 2: Context of Use Model

Problem-2

2.1 Functional Requirements

1. **First Requirement**
   - **Label =** FR1
   - **Difficulty =** Easy
   - **Description =** The function $tan(x)$ allows the user to enter any valid number. For all the inputs, the $tan(x)$ function always provide valid values except for $x = 2\pi + k\pi$. In this circumstance, NaN may be returned.
2. **Second Requirement**
   - **Label =** FR2
   - **Difficulty =** Easy
   - **Description =** When $x = \frac{\pi}{2} + k\pi$, $tan(x)$ returns NaN. $tan(x)$ does not satisfy the domain. The function gives back NaN. There is no need for calculation.
3. **Third Requirement**
   - **Label =** FR3
   - **Difficulty =** Easy
   - **Description =** There is an incorrect format exception thrown when the input is non-numeric. Only numerical values can be passed to $tan(x)$.
4. **Fourth Requirement**

- **Label =** FR4
- **Difficulty =** Medium
- **Description =** When $x \neq \frac{\pi}{2} + k\pi$, $tan(x)$ returns the calculated value. As a result, the computed value will be returned.

2.2 Assumptions

1. Mathematically, the domain of a tangent function can be defined as a set of all possible real numbers. Hence, a real number needs to be entered as input.
2. Based on the JAVA programming language, we assume the inputs fall within the acceptable numerical range. The range of user input should be between - 1.79769313486231570E+308 and + 1.79769313486231570E+308, depending on the data type chosen.
3. In order to comply with the restrictions imposed by the programming language, only 16 decimal places are taken into account and those beyond 16 are removed automatically.
4. For the purpose of the project, the value of pi is assumed to be 3.14159265358979323846 [**7**].
5. Due to the restrictions imposed on data types by programming languages, output values are also removed after 16 decimal digits.

Problem-3

3.1 Mind Map Tangent Calculator

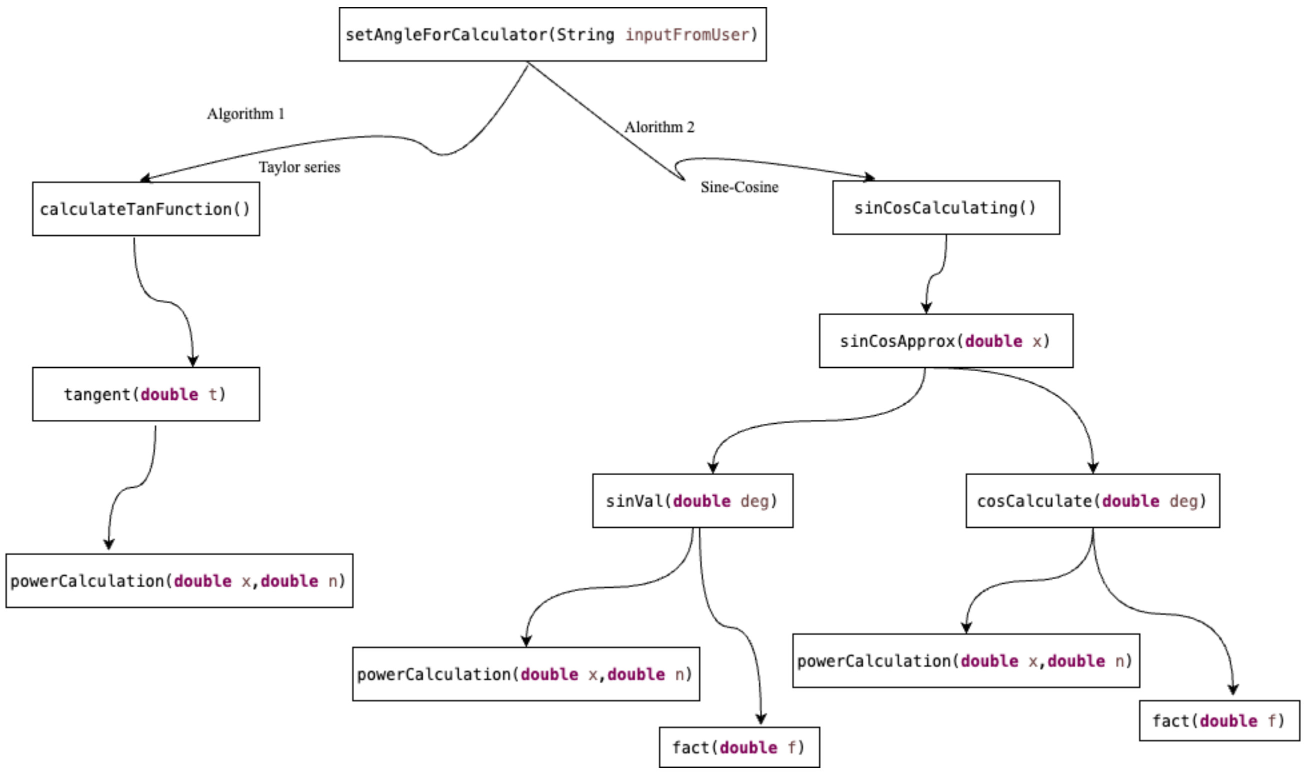

Figure 3: Mind map for Pseudocode Format

3.2 Algorithm-1 [8][9][10][11]

In a Taylor series, all differentials of a function f(x) are present in the region of a point as long as the function is continuous and all its differentials are present. In order to make a series of derivatives from the function, we need to know its order. It is represented as :

$\sum_{n=0}^{\infty} \frac{f^n(a)}{n!}(x-a)^n$

where,
$f^n(a)$ is the function's nth order
The total number is n
The function's focal point is a

Advantages

- When $x >= 0$, the algorithm is applied to all possible real numbers.
- If the functional values and derivatives are recognised at a single point, it is utilised to assess the value of an entire function at each point.
- With this approach, a lot of mathematical proofs are simplified, and the sum of partial series can be used to approximate the entire series, which lowers the complexity of the code.

Disadvantages

- The time it takes is the only drawback. In most circumstances, the only situations in which you should ever make use of it are when you need to find an odd value of a certain function, identify patterns in a proof, or locate integrals that are impossible to solve in any other way.

Reasons to select this Algorithm

- This algorithm is employed for the power flow analysis.
- This can be applied to various optimization strategies, which entails approximating our function as a succession of linear or quadratic forms and iterating on them in order to determine the ideal value.

Pseudocode for Algorithm 1

Calculation of tan(x) using Taylor series expansion

1. start
2. user input
3. validation checking for all conditions
4. if x is outside of the range(0,180) then,
5.   the largest multiple of 180 lower than x is subtracted
6. end if
7. if x is between $(90, 180)$ then,
8. $x \leftarrow x - 180$
9. end if
10. if x is between $(45, 90)$ then,
11. $x1 \leftarrow 90 - x$
12.   return $\frac{1}{tan(x1)}$
13. end if
14. if x is between $(22.5, 45)$ then,
15.   return $\frac{2*tan(\frac{x}{2})}{1-tan^2(\frac{x}{2})}$
16. end if
17. if $x < 22.5$ then,

$z \leftarrow x * \frac{\pi}{180}$

return $z + \frac{z^3}{3} + \frac{2z^5}{15} + \frac{17z^7}{315}$

3.3 Algorithm-2 [12][13]

The population-based metaheuristic technique known as the sine-cosine algorithm is used to solve optimization issues. Individuals within the population move within the search space to approximate the problem, which is done by optimizing. In order to do this, this algorithm utilizes trigonometric sine and cosine functions.

Advantages

- Guarantee population diversity
- Enhance search capabilities
- Increasing the algorithm's performance, particularly when handling challenging optimization tasks

Disadvantages

- The ratio of two independently obtained values are approximates, which is why this algorithm lacks accuracy.
- The calculation of sin(x) and cos(x) functions are necessary for this approach. Hence, takes longer to code and run.

Reasons to select this Algorithm

- Any programming language may easily implement it.
- It saves time by quickly solving optimization issues.

Pseudocode for Algorithm 2

Computing tan(x) as a ratio of sin and cos

1. start
2. user input
3. validation checking
4. if valid, ▷ **For** $sin(x)$,
5. $x1 \leftarrow sin(x)$ ▷ $sin(x)$ is calculated using Taylor's series
6. else
7. exit
8. if valid, ▷ **for** $cos(x)$,
9. $x2 \leftarrow cos(x)$ ▷ $cos(x)$ is calculated using Taylor's series approximation
10. else
11. exit
12. **For Sine-Cosine**,
13. if x == 90,
14. return NaN
15. else
16. return $\frac{x1}{x2}$ ▷ Final tan(x) value

Problem-4

4.1 Programming Style [14][15]

Writing source code for a computer program requires adherence to a set of rules or guidelines, which are contained in a written document known as a programming style. A plugin **"google-java-format"** on IntelliJ version 2021 is available which is used in this program. It helps programmers comprehend and maintain the code more easily and lowers the likelihood that they will make mistakes. Following a style guide eliminates unnecessary uncertainty and speculation.

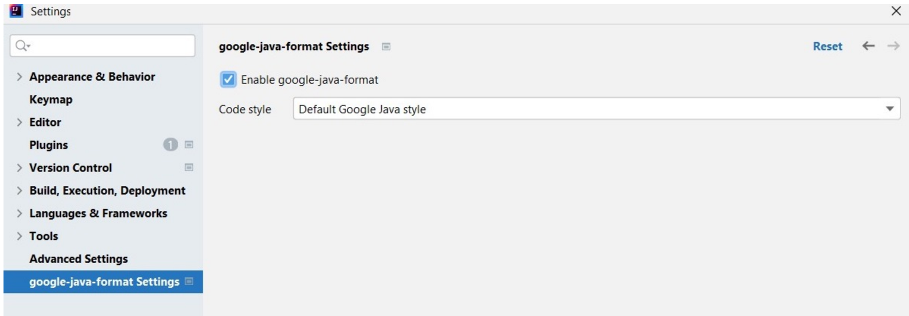

Figure 4: Java Standard Programming Style

4.2 Error handling

Errors occur as a result of improper operations carried out by a user [20]. In order to continue maintaining the code, Java's exception handling mechanism is to ensure that programs run properly even after an exception is encountered and also provides security against system attacks[21]. In this program, there are try-catch blocks that are used to handle exceptions resulting from wrong input and other unforeseen circumstances.

Figure 5: Error Handling for Exception

4.3 Debugger [16][17][18][19]

During the development of a program, programmers use a debugger to test the code in order to ensure that it is not behaving erratically. For implementing my assigned function tan(x), I am using the **Intellij IDE** and the **Intellij Debugger**.

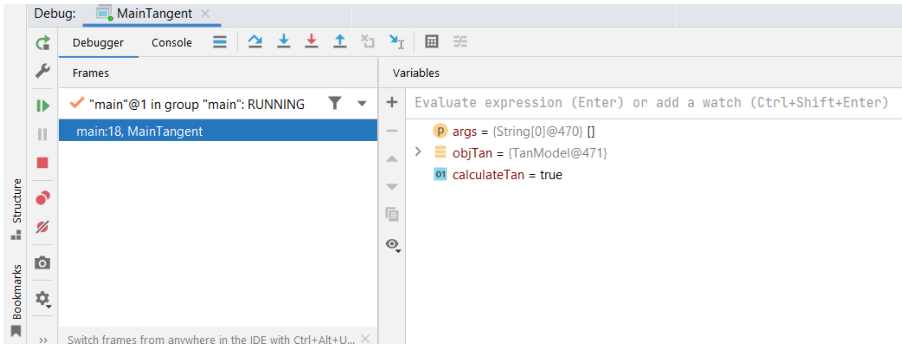

Figure 6: : Debugging the main function

Advantages of Intellij Debugger

- Implement specific breakpoints to inspect the state of the code and determine the value of expressions based on their analysis
- Sessions for debugging can be paused and resumed
- Errors can easily be detected and Memory usage analysis

Disadvantages of Intellij Debugger

- It consumes more system RAM and needs to have a high-end setup in order to run the InteliJ debugger.
- As part of the Intellij IDE for Java, Intellij Debugger is not convenient to use alone.

4.4 Quality Criterion

1. **Space Efficient**
   During the development of this program, the creation variables were reduced in order to make the program space-efficient. There

were not nested loops included in the program to save the space. Google programming style helps in removing unnecessary blank spaces.

2. **Portable**
   This program is platform independent. Without any installer, it runs perfectly.

3. **Maintainable**
   Modularizing the main functionality, handling errors, and user interactions have been done correctly. This facilitates updating of the code and the inclusion of new features easily. The appropriate details are added as comments for better understanding.

4. **Correct**
   To ensure that the program is correct, output is compared to values that have already been computed. Test cases based on Junit are used to ensure accuracy of the implemented application.

5. **Robust**
   Validation is performed on user input to ensure that only numeric values are entered. Data type compatibility is verified beforehand the computation. If the output exceeds the data type's limit, display the appropriate error message to the user.

6. **Time Efficient**
   The software functions effectively as the output is shown in 87 milliseconds after running all the test cases.

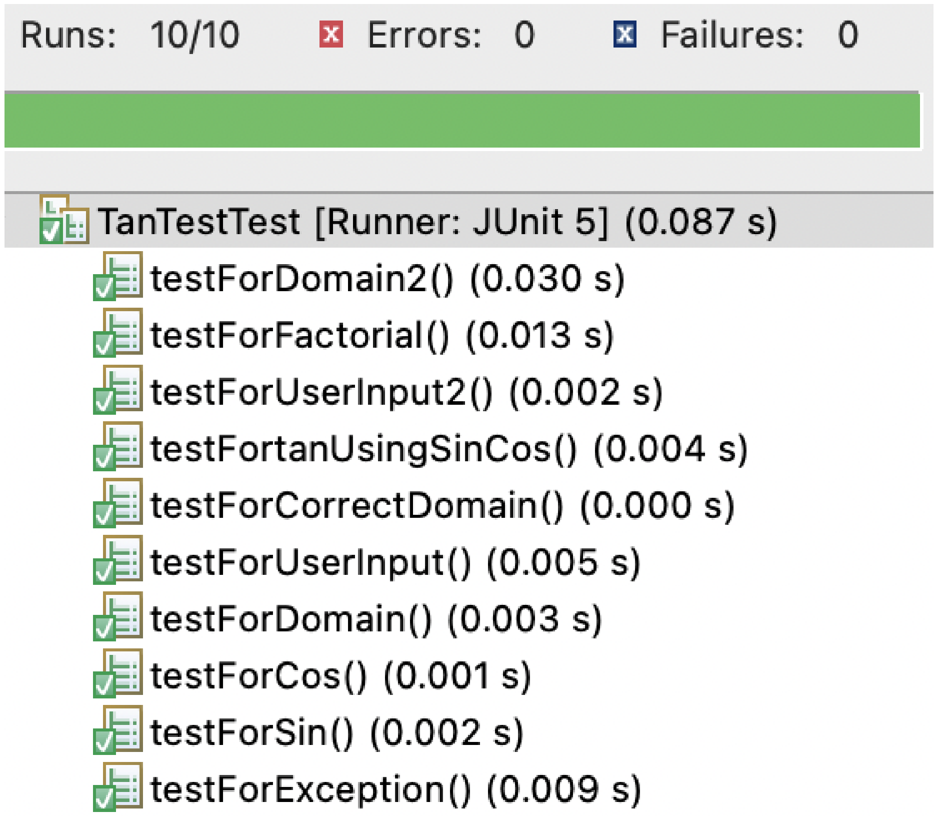

Figure 7: Showing Time for JUnit Test Cases

7. **Usable**

   The user is presented with a text-based user interface that is incredibly straightforward and clear. In addition, the user can easily understand the results and error messages due to their clarity.

4.5 Checkstyle [22]

CheckStyle is a tool that examines the source code to see if it adheres to a specified coding style. The document's readability is enhanced with the aid of this tool.

Advantages

- Easily set up for continuous integration
- operates effectively and enables the configuration of user-defined rules as well as making the code more understandable by examining for design and formatting flaws

Disadvantages

- Does not examine the code for any design flaws.
- It solely involves code beautifying; There are no logic checks.

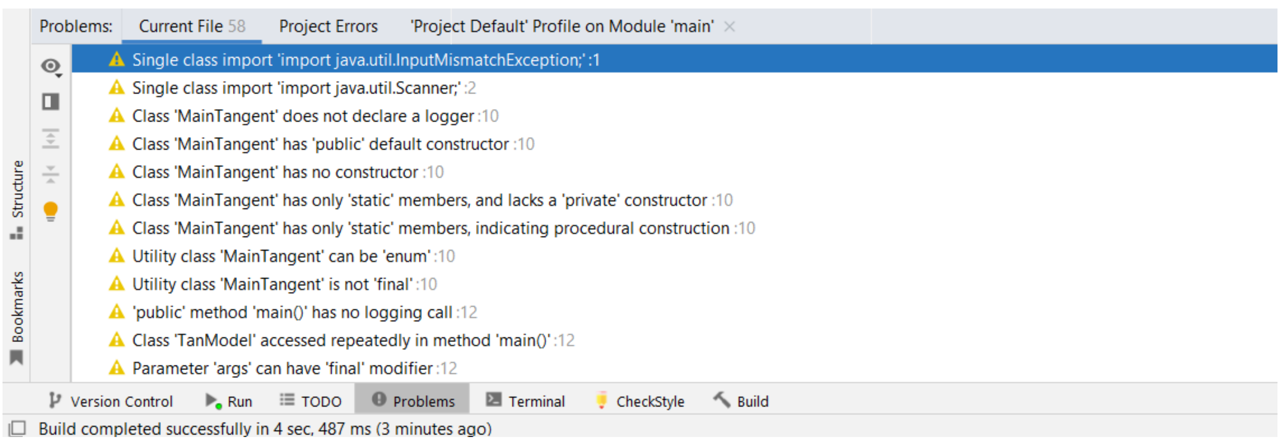

Figure 8: Notification of check style suggestions

Problem-5

5.1 JUnit Standards

In order to write and validated the unit tests of the given function, I used the JUnit framework. In all test cases, the functional requirements were met and the tests were conducted successfully.

5.2 Traceability to Requirements

1. FR1
   JUnit test cases: testForCorrectDomain()
   Description: By performing this test, it is confirmed that this function will produce the appropriate and correct value for the user who enters a valid angle.

2. FR2
   JUnit test cases: testForDomain(), testForDomain2()
   Description: This test verifies that tan(x) returns NaN without computing it if the domain is not satisfied.

3. FR3
   JUnit test cases: testForException()
   Description: This test verifies that an exception is raised for improper format if the input is not numeric.

4. FR4
   JUnit test cases: testForUserInput(), testForUserInput2(), testForFactorial(), testForSin(), testForCos(), testFortanUsingSinCos()

Description: As a result of this test, it has been confirmed that tan(x) returns the computed value if domain is satisfied, including the correct value for it's subordinate function.

References

[1] https://www.cuemath.com/trigonometry/tangent-function/

[2] https://www.onlinemathlearning.com/tangent-function.html

[3] https://www.cuemath.com/questions/is-tangent-even-or-odd/

[4] https://mathworld.wolfram.com/Tangent.html

[5] https://www.cuemath.com/trigonometry/domain-and-range-of-trigonometric-functions/

[6] https://www.rapidtables.com/math/trigonometry/tan.html#rules

[7] https://en.wikipedia.org/wiki/Pi

[8] https://en.wikipedia.org/wiki/Taylor_series

[9] https://www.allmath.com/taylor-series-calculator.php#what-is-the-taylor-series

[10] https://www.vedantu.com/formula/taylor-series-formula

[11] https://www.quora.com/What-are-the-advantages-and-disadvan

[12] https://www.baeldung.com/cs/sine-cosine-algorithm

[13] https://www.hindawi.com/journals/mpe/2020/8184254/

[14] https://www.monash.edu/it/current-students/resources-and-support/style-guide/programming-styles

[15] https://medium.com/level-up-web/what-is-a-programming-style-guide-and-why-should-you-care-

[16] https://www.techopedia.com/definition/597/debugger

[17] https://www.javatpoint.com/intellij-idea-debugging#:~:text=IntelliJ

[18] https://blog.jetbrains.com/idea/2020/05/debugger-basics-in-intellij-idea/

[19] https://blog.gceasy.io/2018/11/05/memory-efficient-eclipse-or-intellij/

[20] https://www.geeksforgeeks.org/types-of-errors-in-java-with-examples/

[21] https://stackoverflow.com/questions/368139/why-is-error-handling-important

[22] https://en.wikipedia.org/wiki/Checkstyle