

Function:-  $\text{Arccos}(x)$

Deliverable- 2

Problem- 4,6

Birva Shah (Student ID: 40070973)

29 July 2019

**Github link:** [https://github.com/BirvaShah/SOEN6011\\_Project](https://github.com/BirvaShah/SOEN6011_Project)

## 1 Changes from D1 to D2

---

**Algorithm 1** Calculating:  $\text{arccos}(x)$  using Iterative Algorithm

---

**function** *CHECKDOMAIN*(*input*)

*in* : user input value

*out* : boolean value

1. *if*(*input*  $\leq$  1.0000 & *input*  $\geq$  -1.0000)

    return *true*

*else*

        return *false*

**function** *CONVERTTODEGREE*(*ansInRad*)

*in* : value in radians

*out* : value in degrees

1. *ansInDeg*  $\leftarrow$  *ansInRad* \* (180/ $\pi$ ())

2. return *ansInDeg*

**function** *PI*()

1. *value*  $\leftarrow$  4

2. *flag*  $\leftarrow$  *false*

3. *i*  $\leftarrow$  3

4. for *i* < 999999

*if*(*flag* = *true*)

*value*  $\leftarrow$  *value* + (4.0/*i*)

*else*

*value*  $\leftarrow$  *value* - (4.0/*i*)

*flag*  $\leftarrow$  (!*flag*)

5. return *value*

---

(**Comment:** I tried to include whole algorithm again but it was not aligning with the changes in bullet points I am mentioning in section 1. So I have just mentioned the changes made in the algorithm to avoid any sort of confusion.)

- Added two more functions **convertToDegree** and **checkDomain** to the pseudo-code of iterative algorithm to calculate  $\text{Arccos}(x)$ .
- **convertToDegree** is an additional function to convert radian values to degrees. I have added this function to support usability. User can see output in both radians and degrees.
- **checkDomain** is a function that is called before the actual  $\text{Arccos}$  function. This function will check if the input value is within the domain range or not i.e between -1 to 1 inclusive.
- **Pi** function is modified to get more precise and accurate value of  $\pi$ .

## 2 Debugger

Debugging is a process that allows us to run a program in an interactive way while watching the source code and its variables at the execution time. It has features like adding breakpoints and watchpoints that are used to stop the execution at any point of time while running a program and investigate on variables, change the values as required.

To implement the function, Eclipse IDE has been used. Eclipse allows our program to run in debug mode. It has in-built default Java debugger under JDT (Java Development Tools) project. It provides features to control the execution via in-built debug commands. So I have used Eclipse default Java Debugger to check for the bugs in my function.

### 2.1 Advantages of Debugger

- It is the ideal way to investigate on problems with our code.
- It can continue the execution of already running program after making changes.
- It has options to enter the internal functional steps or to skip them.
- It can stick to the useful information of data structures for easy interpretation.
- It minimizes useless information.
- The main **disadvantage** of debugger is that the program is not executing in the real-time so may not expose all real-time problems.

### 2.2 Disadvantages of Debugger

- The main **disadvantage** of debugger is that the program is not executing in the real-time so may not expose all real-time problems.

## 3 Quality Attributes

### 3.1 Correctness

The implemented  $\text{Arccos}(x)$  function using **Iterative Algorithm** shows correct result according to the requirements specified. Moreover, test cases are made such that it covers all boundary values and valid/invalid values and gives the expected output.

### 3.2 Efficiency

The function implementation code is efficient in terms of testing. It takes approx. 1.683 seconds to test all the supporting functions and the main function. Moreover, it takes approx. 0.12 seconds to run the code after user enters the real value for calculation of  $\text{arccos}(x)$ .

### 3.3 Maintainability

As standard coding style and javadoc is used in program, maintainability becomes easy. Also the code is divided into separate functions to understand the significance of each and every function in program.

### 3.4 Robustness

Program implemented is robust enough to handle large input or even infinite value. It gives the output according to the conditions applied in the function.

### 3.5 Usability

Function is implemented in such a way that it gives output in two ways. One is in radians and other one in degrees. This way, the function has become more usable for the users to get the clear idea of values they get.

## 4 Checkstyle

Checkstyle is a development tool for programmers to write Java code that adheres to a coding standard. It automatically inspects the Java code if it follows the coding standard. Eclipse has plugin Checkstyle Plugin (eclipse-cs) that integrates the well-known source code analyzer Checkstyle into the Eclipse IDE.

My arccos function .java file has **0 violations** for checkstyle.

### 4.1 Advantages of Checkstyle

- There is portability between IDEs. We can enforce consistency while using checkstyle in different IDEs.
- It is easier to integrate with external tools as it is standalone framework.
- We can add our custom rule in checkstyle.
- It provides good programming practices that improves the quality, readability and re-usability of code. It also reduce the cost.

### 4.2 Disadvantages of Checkstyle

- Checkstyle do not confirm on correctness or completeness of the code.

## 5 Unit Testing

I have used **JUnit-5** testing framework as it is easily available on Eclipse Marketplace. I have written six test cases to check different styles of input and output. The execution time of running all test cases successfully is 1.706 seconds approximately.

## References

- [1] Center for statistical genetics,  
<https://genome.sph.umich.edu/wiki/Debuggers>
- [2] Eclipse Marketplace,  
<https://marketplace.eclipse.org/content/checkstyle-pluggroup-details>
- [3] Vogella,  
<https://www.vogella.com/tutorials/Checkstyle/article.html>
- [4] Developer.com  
<https://www.developer.com/java/other/article.php/2221711/Debugging-a-Java-Program-with-Eclipse.htm>
- [5] Wikipedia,  
<https://en.m.wikipedia.org/wiki/Checkstyle>