Software Engineering Process **Amit Sachdeva**
Topic: Deliverable 2 **40084627**
Prof. P. Kamthan Due Date: 26/July/2019

Gamma Function

Problem 4: Explicit Efforts

1 Correctness and efficient

Gamma Function is basically a factorial function of real values. I have used

$$\Gamma(x) = \int_0^\infty s^{x-1}e^{-s}ds \ \forall \ Re(x) > 0$$

I have implemented it using the core method of integration using graph addition by dividing my function's graph into small chunks and use formula of trapezium to add all chunks. It leads to good accuracy as tested and compared with proved values. Explicit efforts is made to judge the infinity value in reality, and maintain boundaries conditions which can even lead to null values. I have also kept in mind the small decimal values which can range from 0 to nearly $1E-2$ which is so small but are so important for small gamma value results. Apart from this, I have not used any inbuilt math function rather created my own by using **taylor series** which also plus point. So, every boundary condition and using core method of integration lead to high accuracy.

2 Maintainable

Every project need proper maintenance at regular interval of time. So, I have used a MVC design patterns which divide my code in different classes. So I have reduced the dependency of each other class, and any developer want to make changes in functionality of the function, then he/she should focus only on model class. Similarly, If anyone wants to change the view, then he/she should focus only view class. Along with that, proper coding standards is used which reduced the complexity of functions inside the class and proper division of work between the functions. No function is greater than 25 lines of code as defined standards. So, by implementation of design pattern, and proper use of coding standards, maintenance efforts have reduced.

3 Robust

In this project, proper handling of errors inside the models is done. For example if user enters negative value, it returns **negative input error** exception on UI but functionality of code does not hinder.

$\forall \quad value < 0 -> ERROR = "NEGATIVE\ INPUT\ ERROR"$

$\forall \quad value > 109 -> ERROR = "INFINITY"$

$\forall \quad value\ LIKE\ "any\ random\ string" -> ERROR = "WRONG\ INPUT\ ERROR"$

I have used proper history management where user can check what inputs he/she had entered earlier, and what are results which enhances user interaction. So proper checking of errors and using proper exception handling where required so that user interaction does not interrupt.

4 Usability

This project is used by scientist and researchers who indirectly or directly using this function. So, I have created proper **User Interface UI** for them as they don't know how to use command line. I have added functionality like history to check their input history. I have also kept proper error handling and informing user on UI.This lead to increase the usability of any user.