

HACETTEPE UNIVERSITY  
DEPARTMENT OF COMPUTER ENGINEERING  
BBM 104-FIRST ASSIGNMENT REPORT



Name : Buğra

Surname : Tuncer

Number : 21527395

e-Mail : [bugratuncerr@gmail.com](mailto:bugratuncerr@gmail.com)

Assignment No : 1

Programming Language : Java

- **Aim (Description of the Program)**

Shortly, this program does three operation:

- 1-) Integral computation via middle Riemann sum with in a range.
- 2-) Numerical approximation of  $\operatorname{arcsinh}(x)$  function by McLaurin series.
- 3-) Finding Armstrong(narsist) numbers within a range.

- **Software Usage**

I used Eclipse Neon IDE for write this software. This software gets an input file as an argument, reads it then, do some operations and write an output to the console.

- **Solution**

I started with reading files from an input file. We created only one class called "HelloJava" in this assignment. I write six methods to do operations as i mentioned before in the description of the program inside this class. To make this clear, i will explain what my methods do briefly.

1-) factorialFunc : I implemented a factorial method, by the help of recursive functions. It returns factorial value of the number.

2-) fnc1 and fnc2 : These are simple methods which get and double value and returning value of it with a mathematical formula.

3-) fnc3 : I implemented this code to calculate  $\text{arcsinh}(x)$  by the help of McLaurin series. In this assignment we determine the upper value "30".

McLaurin Series for Approximation of  $\text{Arcsinh}(x)$  :

$$\text{arcsinh}(x) = \sum_{n=0}^{\infty} \frac{(-1)^n (2n)!}{4^n (n!)^2 (2n+1)} x^{2n+1} \text{ for } |x| < 1$$

4-) functionSolve : In this method, i implemented a middle Rieman sum for calculating fnc1, fnc2, fnc3 equations' integrals.

Riemann Sum : "The sum is calculated by dividing the region up into rectangles that together form a region that is similar to the region being measured, then calculating the area for each of these shapes, and finally adding all of these small areas together".

5-) armstrongNumbers : In this method i implemented a code for find all Armstrong numbers 1 to n. It does the operations for given digit numbers.

Armstrong Numbers : "Armstrong Number is a number that is the sum of its own digits each raised to the power of the number of digits."

6-) main : My main method reads input file. Then according the input file's arguments it does operations within the methods. Lastly, it will write an output to the console.

## • Personal Comments

### 1-) Difficulties

In this assignment i have faced with many problems. Firstly, i can't truly adopt Java's syntax and logic. In addition to this, there are so many differences in List methods between Java and Python. I try to understand this first then i noticed ArrayList and methods in Java. Lastly, implementing Armstrong numbers algorithm to my code is a bit hard for me.

## 2-)What I Have Learned From This Assignment?

I have learned many things from this assignment.To explain in substance:

- Java's syntaxes.
- Read files with Scanner method.
- Create methods and how to call them.
- How to use effectively Arrays and ArrayLists.
- String concatenating
- Public,private,void,static concepts.
- Mathematical theorems to calculate integral and arcsinh(x).

### • Sample Inputs and Outputs

#### #Sample 1

IntegrateReimann Func1 -5 12 40	Result: 608.91078125
IntegrateReimann Func1 -5 12 200	Result: 609.1564312500004
IntegrateReimann Func2 -12 17 10	Result: 565.5404384807163
IntegrateReimann Func2 -12 17 100	Result: 568.4083507135516
IntegrateReimann Func3 0.2 0.6 50	Result: 0.15517096022190469
Arcsinh 0.4	Result: 0.3900353327969141
Armstrong 3	Result: 0 1 2 3 4 5 6 7 8 9 153 370 371 407

#### #Sample 2

IntegrateReimann Func1 -8 6 10	Result: 296.37999999999994
IntegrateReimann Func1 -8 6 50	Result: 298.575200000000005
IntegrateReimann Func1 -8 6 100	Result: 298.6438
IntegrateReimann Func1 -8 6 150	Result: 298.6565037037039
IntegrateReimann Func2 -4 13 10	Result: 384.1318860305626
IntegrateReimann Func2 -4 13 50	Result: 382.1284469791137
IntegrateReimann Func2 -4 13 100	Result: 382.0504600486257
IntegrateReimann Func2 -4 13 150	Result: 382.0359648793621
IntegrateReimann Func3 -0.1 0.4 90	Result: 0.07398540663088961
Arcsinh 0.7	Result: 0.6527128616017508
Armstrong 4	Result: 0 1 2 3 4 5 6 7 8 9 153 370 371 407 1634 8208 9474