# Laboratory -2: Java Basic

Md. Alamin

ID-21303134

Fall-2024

## **Task-1:**

1. The logical problem arises due to the switch statement. We know switch statement only works with a specific value (ex: 90, 80) matched with “case”. It’s not working with range like 80 to 90 or 60 to 70. That is why, if we input 85 instead of direct 90, we will see the output as “F”.
2. The logical solution can be solved by the switch statement. But there we have to change the values of the corresponding grade to check inside the switch case statement. We will take input as according to our question. Then the inputted value will be divided by 10 and the result will work as range 60 to 70 or 80 to 90. **For example**, if you get 90 then 90/10 = 9 equal “A” again, 75/10=7.5 as it is integer, so the decimal part will be removed, actual value will be 7, “C”;

## **Task-2:**

1. In this program, the logical issue arises due to the uses of multiple if condition instead of if-else. The logic set such way; each if will call during the run time of main method. That’s why each if statement check their conditions and if they meet the condition, they print the output.
2. The Grade2.java code has been updated to solve the logical error.
3. We use && (and logical operator) 4 times in the program;
4. We don’t use any conditional operator other than Relational Operators; We used <= (less than or equal to) 1 time; used >= greater than or equal to 4 times, Used < less than 4 times;
5. There is no difference between my && (logical AND) solution and & (bitwise AND) solution; Bitwise and & work correctly. We know, logical and (&&) is used to combine to Boolean expression and result will be true if the both expressions are true but it goes to second expression if the first expression outcome is false; On the other hand, Unlike the logical AND, it always evaluates both sides of the expression, even if the first one is false.
6. We have modified the code and reduced the operator using the **conditional operator**.

## **Task-3: The BMI program has been done with the file name Task3BMI.java**

## **Task-4:**

1. if (x \* x + y \* y <= 1) {

numberOfHits++;

the code segment is checking here the randomly generated numbers is for x + y is equal to 1 or less; thus, we set the radius = 1; so, if randomly number’s summation is less than or equal to 1 it will increment the numberOfHits with one;

1. double pi = 4.0 \* numberOfHits / NUMBER\_OF\_TRIALS;

the logic behind the following statement is geometrical relationship between the circle and the bounding square.

## **Task-5:**

* 1. The first inputted value was **15** and the second value was **55. Elapsed Time = 1100**
  2. The program iterates through all possible divisors (k) starting from 2 up to the smaller of the two numbers (n1 and n2). If both numbers are divisible by k, it updates the gcd value. This way, the last value of k that divides both numbers is the Greatest Common Divisor (GCD).
  3. After updating the code while loop to for loop, the The first inputted value was **15** and the second value was **55. Elapsed Time = 1100; no changes in time;**
  4. After updating the code segment to k <= n1 / 2 && k <= n2 / 2; the The first inputted value was **15** and the second value was **55. Elapsed Time = 900; We can see the changes here according to time;**
  5. In my case, I could not found any error but correct answer for the case of 2 and 5 or 5 and 2; the result was 1. And the time was 500;

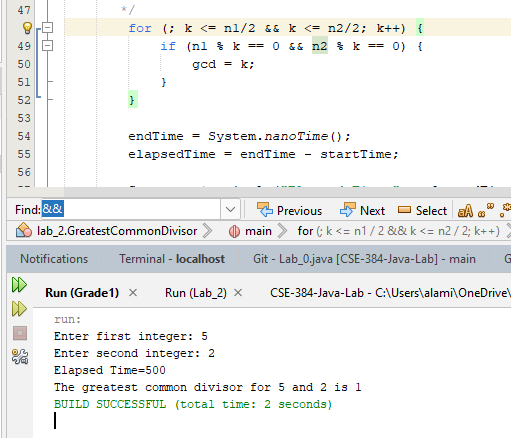


Figure - for the reference of question e;

f) I changed the **k <= n1 / 2 && k <= n2 / 2;** in to **k <= n1 && k <= n2** to see if there is some differences in terms of time. But I see there is no difference with the previous one.

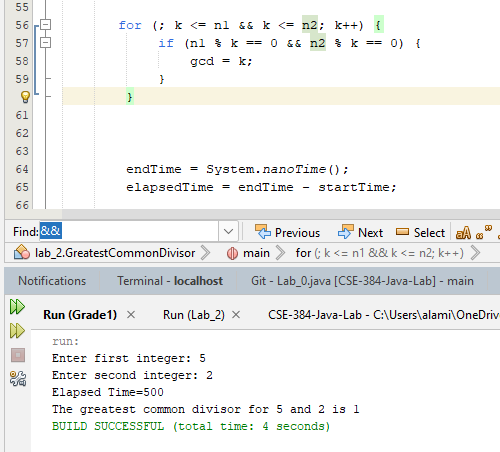


Figure - for the reference of question f

1. After running the GreatestCommonDivisor3.java file, the first inputted value was 5 and the second value was 2. Elapsed Time = 700; We can see the time differences here;
2. Even though the program takes slightly longer than the simpler iteration method for small numbers, but Euclid's method generally scales better with larger numbers.