Muazzam Shaheryar

CSC 3100

Hexadecimal to Binary and Decimal Converter Explanation

This is a very simple convertor really. In the start you are greeted with a menu where you can select what feature you want to use. The menu uses while loop and switch statements to work properly.

**Hexadecimal to Decimal**

In this method you enter your hexadecimal value and it is converted to decimal.

What the method does is that, it first it separates your hexadecimal value into separate characters. A for loop makes sure all alphabet values are capitalized, if not then makes them upper case. Since the Hexadecimal value was stored in a string, it would be separated into chars. Chars use ascii, so you convert from ascii to decimal which is done by removing 48 for numbers (0-9) as 48 is the first number in the ascii code then you multiply it with the base which is 16^x.

For alphabets you do it a little different because you remove 55 instead of 48. 55 is used because even though in ascii alphabets start at 65. This is because in hexadecimal 10 is A (first alphabet), so to achieve this we have to create an added gap of 10. Then you multiply with the base which is 16^x.

**Hexadecimal to binary (without preceding zeros)**

In this method first, you convert the hexadecimal to decimal using the previous method (hexadecimal to decimal) then you convert the method to binary. To do this you get the remainders (obtained by diving the decimal with 2) into one ArrayList and keep diving by until the quotient becomes 0. After which you copy to a String Builder to display it all together then flip it to get it displaying the correct way and convert it to string for easier output.

**Hexadecimal to Binary (with preceding zeros)**

In this method you have 2 arrays which store the binary values from 0 – 15 let’s call this a and another which has hexadecimal values from 0 -15 let’s call this b. A for loop makes sure all alphabet values are capitalized, if not then makes them upper case. You compare your separated hexadecimal values with the b and since both b and a have the same value in position just in different form (a has in binary and b has in hexadecimal), you use the position number from b and add that corresponding binary number to a StringBuilder. After it is all completed the StringBuilder string is converted to string and outputted.