Abstract

The programs Converter is designed to do multiple conversions. It converts from binary to decimal. decimal to binary, decimal to hexadecimal, hexadecimal to decimal, hexadecimal to binary and binary to hexadecimal

Java e project

Converter

Table of Contents

[CONTRIBUTORS 2](#_Toc13747296)

[Acknowledgement 2](#_Toc13747297)

[Dedication 2](#_Toc13747298)

[Introduction 3](#_Toc13747299)

[Decimal to binary 3](#_Toc13747300)

[Binary to decimal 4](#_Toc13747301)

[Decimal to hexadecimal 4](#_Toc13747302)

[Hexadecimal to decimal 4](#_Toc13747303)

[Binary to hexadecimal 5](#_Toc13747304)

[Hexadecimal to binary 5](#_Toc13747305)

[Binary Addition 6](#_Toc13747306)

[Conclusion 6](#_Toc13747307)

# CONTRIBUTORS

# Allison T Magaji

# Sule Philip

# Zainab Ahmed Ibeto

# Munir Ahmed Ibeto

# Williams Osita Nwokeafor

# Surrayyah Hamza Sambo

# Acknowledgement

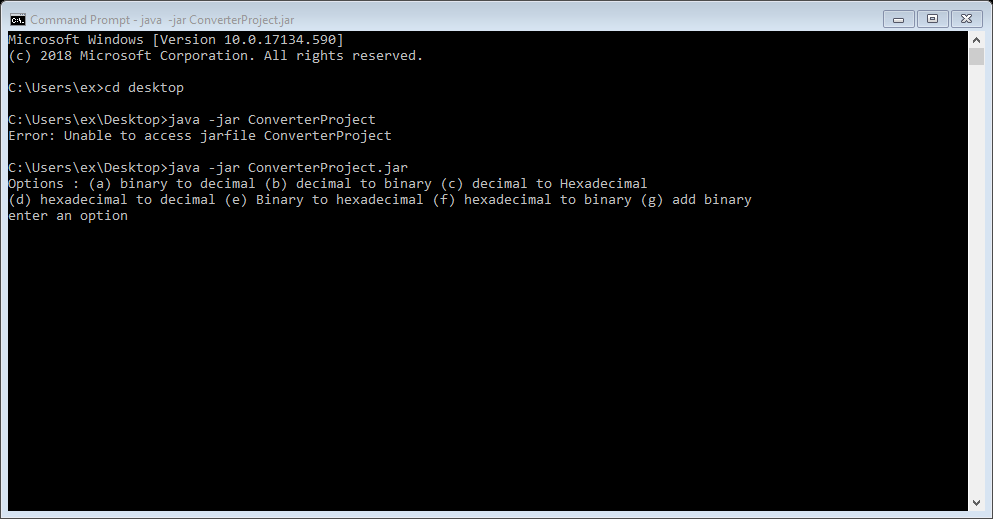
We acknowledge the support of our faculty who has been a helpful guide throughout the process. We also acknowledge the countless hours of work that the selfless developers at JAVA and have put into making source code development as painless as possible.

# Dedication

This project is dedicated to our hard work and everybody that helped in the process of creating this binary project, every single person who contributed like Philip, Allison and other people.

# Introduction

The programs Converter is designed to do multiple conversions. It converts from binary to decimal. decimal to binary, decimal to hexadecimal, hexadecimal to decimal, hexadecimal to binary and binary to hexadecimal. It also performs addition between two binary numbers. Each of these operations were broken down into separate function and then combined together to form the Converter program

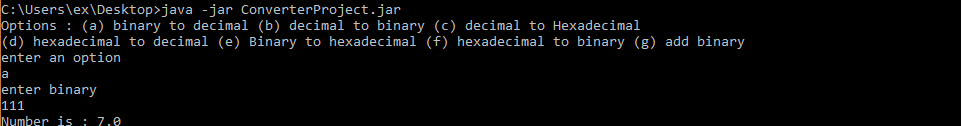


# Decimal to binary

This function takes in an in t in the parameters then has a variable r for storing remainder and a string binary which would be the final output. A while loop is set to keep running until the number(num) turns to zero. Inside the while loop r stores the remainder of the number divisible by 2., the string binary attaches the remainder to its front and the number new value is itself divided by 2. Once the while loop breaks the value of binary is printed out.

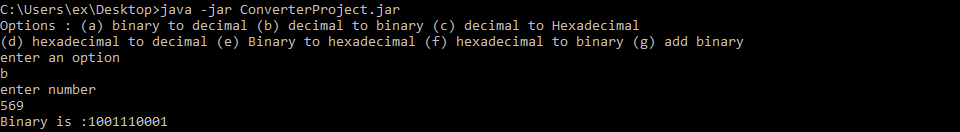
# Binary to decimal

This function takes in a string num and uses a for loop to go through each character and each character at position I is tines by 2 raised to the power of inverted position and added to an integer number. For example, using the String “1001”, in position 0 the character 1 will be times by 2 raised to the power of the inverted position which would be the last position ‘3’.



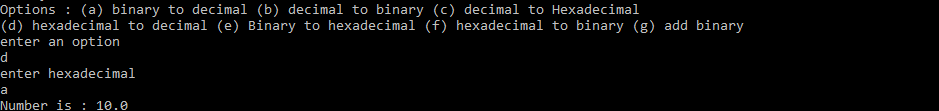
# Decimal to hexadecimal

The function takes a number in the parameter, it has an integer r where it stores the remainder. Then a String Hexadecimal which stores all the concatenations. A while loop is then implemented until num is 0. R stores the number divisible by 16 then comes switch statement that depending on what r, it will add a string to the beginning of hexadecimal. The rules are set that for if r is 10, strings an A. if 11 stores B, if 12 stores C, if 13, stores D, if 14 stores E, if 15 Stores F, then for everything else it just stores the string of that number. The new value of num becomes num divided by 16. When the while loop breaks, hexadecimal is printed out.



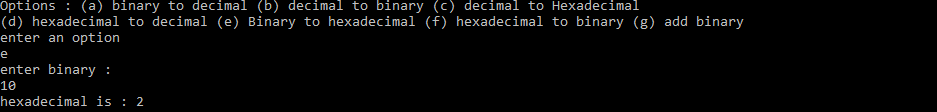
# Hexadecimal to decimal

This function takes a string num and has a double number where it stores the sum of everything. A for loop goes through each character of num. there is a switch statement set up for every character. For A the number is 10, for B the number is 11, for C the number is 12, for D the number is 13, for E the number is 14 for F the number is 15. For everything else its just, the number. Basically the int versions of the character is times by 16 raised to the power of (n-1 – iterator) with n starting at the end the answer is then added to the value of the number int. once the loop finishes the number will be printed.



Binary to hexadecimal.

This function takes a String num and checks if it is divisible by 4 without remainder. If it has that remainder will be the number of zeros to add to the beginning of num using a for loop. After there is a for loop that goes through the whole string of num. In the loop a substring of temp is stored from position of the iterator to the next four positions. This iterator I increases by four. then inside there is a second for loop that goes through tmp and adds the sum of each characters value times two raised to the power of (length – 1) - iterator j and stores it in sum.. when the second for loop is done, there is a switch statement depending on the value of sum a string would be added to hexadecimal (string storing the hexadecimal values). If sum is 10 it will add A, if sum is B it will add 11, if the sum is 12 it will add C, If the sum is 13 it will add D, if the sum is !4 it will add E, if the sum is 15 it will add F and for the rest it will just add the number string.. Once the first for loop is done iterating hexadecimal will be printed out.

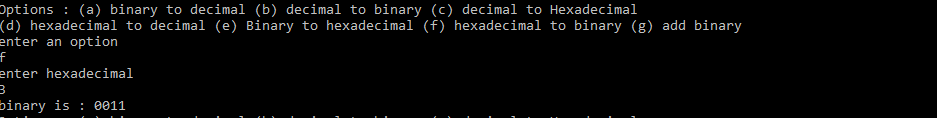


# Hexadecimal to binary

This function takes a string hexadecimal and uses a for loop to iterate through all its characters. And the character will be stored in tmp which then a switch statement checks to see what the character is. Depending on the character, a string will be add to a string binary. Below are the rules setup:

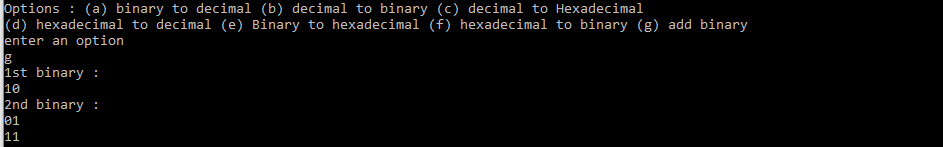
If sum is 0 binary = binary+ "0000 if sum is 1 binary = binary+ "000 if sum is 2 binary = binary+ "0010" if sum is 3 binary = binary+ "0011"if sum is 4 binary = binary+ "0100" if sum is 5 binary = binary+ "0101";if sum is 6 binary = binary+ "0110";if sum is 7 binary = binary+ "0111"; if sum is 8 binary = binary+ "1000";if sum is 9 binary = binary+ "1001";if sum is A binary = binary+ "1010";if sum is B binary = binary+ "1011";if sum is C binary = binary+ "1100";if sum is D binary = binary+ "1101";if sum is E binary = binary+ "1110";if sum is F binary = binary+ "1111";

At the end of the for loop binary will be printed out.



# Binary Addition

This function will take 2 string a and b. Pos1 one is set at the size of a – 1 and pos2 is set as the size of b -1. There is a while loop set up that if both pos1 and pos2 are greater or equal to zero then both strings are iterated through with there own iterator. If a at pos1 = 0 and b at pos2 = 0 and c = 0 then “0” is added to sum which stores the sum value of the added binaries. If a at pos1 = 1 and b at pos2 = 0 and c = 0 then “0” . If a at pos1 = 0 and b at pos2 = 1 and c = 0 then “0”. If a at pos1 = 1 and b at pos2 = 1 and c = 0 then “0”. If a at pos1 = 0 and b at pos2 = 0 and c = 1 then “0”. If a at pos1 = 1 and b at pos2 = 0 and c = 1 then “0”. If a at pos1 = 0 and b at pos2 = 1 and c = 1 then “0” is. If a at pos1 = 1 and b at pos2 = 1 and c = 1 then “1”. Then both positions are subtracted by 1. After the loop finishes iterating the there. There is another while loop that checks if a has any characters left that have not been iterated through. If there is it goes through the rest and applies the rules above. Then after that there is another while loop that does the same for b. then after that if c is equal to 1 it adds it to sum. The sum is then printed out



# Conclusion

What we set out to develop was a well-polished user experience and to attract the user to the site. Also, one of our main goals was to accurately represent the cars and to not under-sell our skills.

In this regard, we have definitely accomplished our objectives and overcome the obstacles with very simple solutions. We've designed not just a visual experience, but also an emotional experience for the user. In the process, we've learned so much and gained so much experience.