**Project Name**

Ultimate Converter

**Description**

This program will build upon the already completed binary calculator, allowing the user to convert from the following numbering systems: binary to decimal, decimal to binary, decimal to octal, octal to decimal, hexadecimal to decimal, decimal to hexadecimal, binary to octal, octal to binary, binary to hexadecimal, and hexadecimal to binary.

**Problem You Are Trying to Solve**

Before any student is to enter the field of Computer Science or Computer Engineering, he/she should be able to understand the fundamental features of how computers work the way that they do, and how a combination of electricity and hardware can result in a powerful and fully functioning machine. At the basis of this subject matter lies the state of *binary*, which literally translates to “of two states.” If a computer is to receive electrical flow from one location, it is denoted as a 1. If there is an absence of electricity, it is denoted as a 0. It is through this basic concept that the simplest of circuits can be designed. By understanding this idea, students can therefore have a better understanding of what a circuit really is, and why it works the way it does.

However, there is another concept that lies within this idea that is equally as important. When it comes to remembering numbers, computers are not advanced or capable enough to understand our traditional decimal system. There are only two things a computer can know, and that is “on” or “off.” So, when it comes time to store information into its memory, a computer must convert the data into something that it can understand, in order to be able to save it successfully. This is where we see once again the use of binary.

That being said, binary is not ideal. It is a system that does not use a human’s time efficiently, and it is tedious to understand. That is where the usage of octal and hexadecimal come into play. By understanding these numbering systems, binary can be quickly converted and thus understood, and is extremely beneficial for understanding why a bug might occur in computer code or why a certain piece of hardware might seem to be malfunctioning.

**Benefit of Having This Program**

My program aims to help teach these concepts in a simple-to-understand and interactive way. With the combination of arrays, as well as multiple body functions, the user will be able to navigate from one part of the program to another in an integrated and efficient manner. Some features that might be included are things such as the program making use of diagrams in the forms of images that would be accessed through the downloadable OpenCV library, or to allow a user to save his/her inputs and outputs through the use of a text file through the already included fstream library.

**Future Evolution of the Program**

Eventually, this program can evolve from simply a console application to a full featured window application with nice visuals and graphics. The code itself can evolve into a more professional state making the use of classes and objects, as well as make better use of things like pointers, to better help in avoiding errors, crashes, or general failures. The program may also make use of more downloadable libraries that allow for more functionality and better/cleaner overall code.

**Development and Delivery Schedule**

The overall skeleton of the programs functionality can be completed within 2-3 weeks, with the more advanced features (such as writing to a file or opening image diagrams) to be added later on. However, considering that these extra features are subject matters not taught within the classroom setting, these features are subject to malfunctioning/failure, and will therefore only be considered as additional features that may be included in the program time permitting. Assuming all features included, the program should be completed within 1-2 months’ time.