**CS 2060 Assign03 Problem Solving**

Your design notebook is a problem-solving tool. They don’t need to look nice. They’re not for presentations or reports. A good notebook will be full of false starts, rough sketches, alternative designs, questions that need answers, research notes, insights, and reflections.

# Class Activity 03

**Purpose:** To see how you approach a problem before you write code.

Start your design notebook for this problem:

Print the Decimal Equivalent of a Binary Number. Input an integer (5 digits or fewer) containing only 0s and 1s (i.e., a “binary” integer) and print its decimal equivalent.

* Complete **Part 1 and Part 2**
* Upload this document to canvas for Class Activity 03 **before the end of class**.

|  |
| --- |
| **Part 1: Problem Statements and Understandings (*required*)** |
|  |
| In your own words, what the assignment is asking you to do.  The assignment is asking us to to take a 5 digit binary value, then convert it into a decimal number  Maybe including an example will help. |
| For example, do you know how to convert a binary number to decimal?  **DO** know   * I know how the conversion from binary to decimal works mathematically. * I know how to store the proper values in the integer.   DO **NOT** understand   * I do not know how to convert the binary conversion arithmetic to code.   . |
| Sources you used (Lecture notes, Textbook, Google, Instructor) and what you learned.  For example:   * How do you convert a binary number to a decimal? * How do you find each bit when the computer reads the number as a decimal value?   + 101 means the first bit is a 1 the second bit is a 0 and the third bit is a 1 and not one hundred one in decimal.   --Lecture notes, past instructors, textbook  --you could split the number into an array of bytes |

|  |
| --- |
| **Part 2: Test Data (*required*)** |
| Test Data helps the developers to find the problem during fixes. Test Data may be used in a confirmatory way, typically to verify that a given set of input to a given function produces some expected result   * Check that outputs are derived correctly. * Varied for different scenarios in testing(valid/invalid) * Help developers find problems * Verify input to a given function produces expected result. **(Not necessary for this assignment)** * See if program can handle unexpected inputs. **(Not necessary for this assignment)**   **Update the test cases as you go through this process and then write the code.** |
| Create a list of test data and expected results.  00101  11010  11111  00000  11010  1101  1010  110  11  11200 |
| Explain why you chose this for your test data and what you learned from the results.  ---I chose those test cases because It has the expected data, and the unexpected data, so I can work towards making my program more robust and no as hard-coded |

|  |
| --- |
| **Part 3: Design Sketch (*required*)** |
| Start drawing pictures and diagrams on paper   * Even if you don’t understand the problem fully, or don’t have a solution, start sketching fragments, pieces of the puzzle. * Use the simplest possible notation such as draw boxes, lines, arrows, and notes. * Show the major components and the flow. * Don’t invest time in drawing formal diagrams, such as UML, just get the ideas flowing. * No need for the sketches to look nice – but they must be clear and legible!   Scan the pages and include the images in your OneNote notebook. There are several apps available to scan your design notes. For example, [CamScanner](https://www.camscanner.com/) is an app that is easy to use and provides a free “basic” account. |
| Asketch showing your brainstorming and the **steps involved** in the problem, flow of data, etc. |

|  |
| --- |
| **Part 4: Pseudocode (*required*)** |
| Pseudocode is similar to everyday English to help you think about the problem and help you find a solution.  Write the pseudocode before you write the code. This will help you determine the actions and decisions to make writing the code easier. |
| Write pseudocode to show how main() works (or user defined functions if you create them) |

|  |
| --- |
| **Part 5: Reflection (*required*)** |
| Ideas of things to reflect on.   * What did you learn? * What problems did you encounter and how did you solve them? * What lessons did you learn that you would want to remember for the future? * What concept did this problem help you understand? * What other functionality could be added to the program? |
| Write at least one paragraph reflecting on your work. |