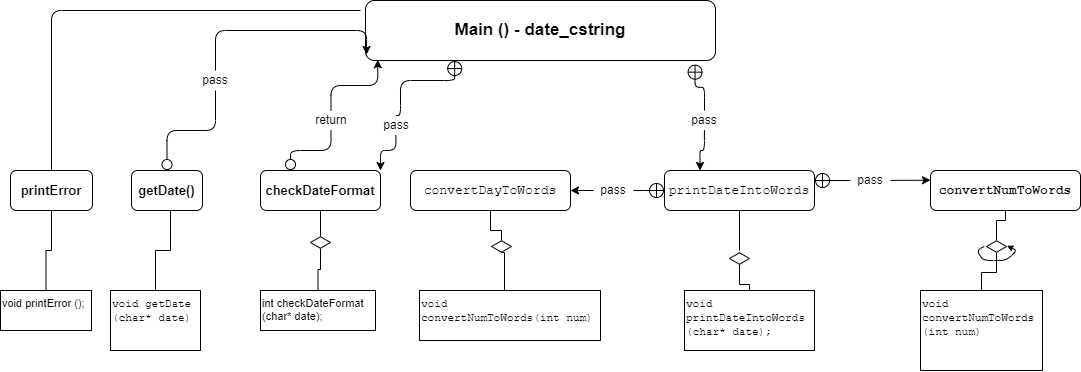
**CSC 250 – Program Design Document**

**John Akujobi**

**Structure Chart**



**Constants**

1. const char\* const ones[] - The array of strings for the ones place
   * {"", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine"};
2. const char\* const oneths[] - The array of strings for the ones place
   * {"", "first", "second", "third", "fourth", "fifth", "sixth", "seventh", "eighth", "ninth"};
3. const char\* const tens[] - The array of strings for the tens place
   * {"", "", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};
4. const char\* const elevenToNineteen[] - The array of strings for the teens
   * {"Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"};
5. const char\* const months[] - The array of strings for the months
   * {"", "January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"};

**Variables in Main function**

1. char date[11] - C-string to store the date
2. int datecheck - Variable to store the result of the checkDate function
   * (0 = valid, 1 = invalid)
3. char choice - Variable to store the user's choice to continue or not
   * (y = yes, n = no)

**Function Design (give the prototype and a short description for each function)**

**Functions Design (give the prototype and a short description for each function**

1. getDate() - Function to get the input date
   * void getDate(char\* date);
   * Gets the input date from the user and stores it in a C-string.
   * Takes in char pointer “date” as input and returns nothing.
2. printError() - Function to print the error message
   * void printError ();
   * Prints an error message to the console when the input date is not in the correct format.
   * It takes no input and returns nothing.
3. checkDateFormat() - Function to check the format of the input date
   * int checkDateFormat(char\* date);
   * Checks if the input date is in the correct format of MM/DD/YYYY.
   * Takes in char pointer “date” representing the input date as input and returns an integer.
   * If the input date is in the correct format, it returns 0. Otherwise, it calls printError() to print an error message and returns 1.
4. printDateIntoWords() - Function to convert date to words
   * void printDateIntoWords(char\* date);
   * Converts the input date to its corresponding word form and prints it to the terminal
   * Takes in char pointer “date” representing the input date as input and returns nothing.
   * It first checks the input date is in the correct format using checkDateFormat(). Then uses convertNumToWords() and convertDayToWords() to convert the year and day to words, respectively, and months array to convert the month to words.
5. convertNumToWords() - Function to convert number to words
   * void convertNumToWords(int num);
   * Converts an integer between 0 and 9999 to its corresponding word form.
   * It takes an integer input num as input and returns nothing.
6. convertDayToWords() - Function to convert day into words
   * void convertDayToWords(int day);
   * Converts an integer input "day" between 1 and 31 to its corresponding word in the form of first, eighth, twenty third and so on.
   * It takes an integer input "day" as input and returns nothing.

**Time Estimate**

|  |  |  |
| --- | --- | --- |
|  | **Estimated Time** | **Actual Time** |
| Program Design | **30 mins** | **45 mins** |
| (list each function name and the time required to code each) | **90 mins** | **87 mins** |
| getDate() | **10 mins** | **4 mins** |
| printError() | **5 mins** | **2 mins** |
| checkDateFormat() | **20 mins** | **13 mins** |
| printDateIntoWords() | **20 mins** | **20 mins** |
| convertNumToWords() | **15 mins** | **12 mins** |
| convertDayToWords() | **5 mins** | **4 mins** |
| Main | **15 mins** | **20 mins** |
| Program Test | **1hr** | **45mins** |
| **Total Time** |  |  |