



Objective:

- Focus on the purpose/use of class level information (data/operations).

Task - 1: discussed in class/lecture

<pre>class CMath { public: static float calcPower (int, int); static int calcGCD (int, int); static String toString (long long int);</pre>	<pre>static long long int toInteger (String); //you may add other mathematical functions in the same way };</pre>
---	---

Task - 2:

Design a class called 'Date'. The class should store a date in three integers: month, day, and year. There should be member functions to print the date in the following forms:

Format-1: 12/25/2012

Format-2: December 25, 2012

Format-3: 25-Dec-2012

Demonstrate the class by writing a complete program implementing it.

Your setter functions should make sure following:

- A valid year is between 1500 and 3000
- A valid month is between 1-12
- A valid day can be between 1-31 (according to the respective month)

Make following daysInMonth array as class's private data member to easily know the number of days in each month.

```
static const int daysInMonth[ 13 ] = { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
```

```
class Date
{
    int day;
    int month;
    int year;
    static const int daysInMonth[ 13 ];
    bool isLeapYear () const;
public:
    Date ( );
    Date ( int, int, int );
    void setDate ( int, int, int );
    void setDay ( int );
    void setMonth ( int );
    void setYear ( int );
    int getDay ( ) const;
    int getMonth ( ) const;
    int getYear ( ) const;
    void printFormat1 ( ) const;
    void printFormat2 ( ) const;
    void printFormat3 ( ) const;
    void incDay ( int = 1 );
    void incMonth ( int = 1 );
    void incYear ( int = 1 );
    String getDateInFormat1 ( ) const;
        //if *this object contains day=25, month=12 and year=2012 then it returns a Sting
        //object containing "12/25/2012"
    String getDateInFormat2 ( ) const;
        //if *this object contains day=25, month=12 and year=2012 then it returns a Sting
        //object containing "December 25, 2012"
    String getDateInFormat3 ( ) const;
        //if *this object contains day=25, month=12 and year=2012 then it returns a Sting
        //object containing "25-Dec-2012"
};
```

Note: You are not allowed to use C++ string functions but you are free to use String function wherever needed.



Objective:

- It will help you understand the idiom "Use const wherever possible" and "Principle of least privilege".

Task-1:

An updated version of *String*, which will provide basic functionalities related to strings.

Note: You are not allowed to use any library functions related to strings.

```
class String
{
    char * data;
    int size;
public:
```

String ();	Initializes data to nullptr and size to 0.
String (const char c);	Initializes data with char c
String (const char *);	Initializes the data with received string by allocating memory on heap.
String (const String &);	
~String ();	You know what to do.
void input ();	Takes input from console in *this object.
void shrink();	Resize/shrink the array equal to the length of string pointed by data.
char & at (const int index);	Index: Receives the index for string. Return Value: reference of array location represented by index
const char & at (const int index) const;	
bool isEmpty () const;	Tells whether string is empty or not? An empty string is one which has data==nullptr or data[0]=='\0'. Return Value: return true if string empty otherwise false.
int getLength () const;	Returns length of the string
int getSize () const;	Returns the value in the size attribute.
void display () const;	Prints the string on console
int find (const String & subStr, const int start=0) const;	Find the first occurrence of substring in the calling String object. By default, search starts from 0 index. If found then return the starting position of subStr found otherwise return -1.
void insert (const int index, const String & subStr);	Insert the substring at given index in calling object.
void remove (const int index, const int count=1);	Remove the characters (how many? Given in count) starting from index
int replace (const String & old, const String & newSubStr);	Find all the occurrences of old substring and replace it with new substring. Return the count of occurrences found in calling object.
void trimLeft ();	Removes all the white space characters on the left of string
void trimRight ();	Removes all the white space characters on the right of string
void trim ();	Removes all the white space characters on both left and right sides of string
void makeUpper ();	Change all the alphabets to uppercase
void makeLower ();	Change all the alphabets to lowercase
void reverse ();	It reverses the string stored in the calling object
void resize (int);	You know what to do.
int compare (const String & s2) const;	Compare the calling and receive object string and behave just like strcmp
String left (const int count) ;	Count: The number of characters to extract from calling object from left side Return Value: A String object that contains a copy of the specified range of characters
String right (const int count) ;	



<code>long long int convertToInteger () const;</code>	Converts the integral value stored in calling object to long long int and returns the integral value.
<code>float convertToFloat () const;</code>	Converts the fractional value stored in calling object to float and returns it.
<code>String concatenate (const String & s2) const;</code>	It returns the concatenated result of received and calling object without changing calling object.
<code>void concatEqual (const String & s2.);</code>	It concatenates the received object string with calling object.
<code>void setNumber (const long long int num);</code>	Stores the received number as a string. String s; s.setNumber(-592950); s.display(); //prints -592950 s.setNumber(489); s.display(); //prints 489