**JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY–NOIDA**
**CI311- Object-Oriented Programming**
**Tutorial / LAB – Week 2**

**Absence from the regular non evaluative labs can fetch negative or reduced marks in the successive evaluative labs. So students are hereby notified that they maintain their 100% attendance and do their assignments on regularly basis from day 1and don't skip any of the labs including the non evaluative labs. No excuse will be entertained of the absence from the labs.**

**Based on Week 1 and Week 2 Assignments first Quiz will be held in the third week.**
**(i.e from 25 to 30 Aug)**

## Practice Set-II

**Question 1.** Write a class in C++ representing a one dimensional Array of integer. Design suitable member functions to add, delete a value from array, sort array using insertion sort and search a value in the array using binary search. Operations should invoke at runtime based on user choice. Write a menu driven main to demonstrate all functions of an array class. **Write at least three inputs with which you have tested your class.**

**Question 2.** On Similar pattern, write a String class that can store any length string of characters. It should have functions that imitate C++ library functions related to string.-String Copy, String Reverse, String concatenate, String Compare and Find a Character occurrence in a String. These functions should be designed such that they produce a new resultant String object wherever possible. **Write at least three inputs with which you have tested your class.**

**Question 3:** Write a class test which contains a string() function, which will further display a string at a given row and column on a given VDU page, in a given color. If the page number and the color are not passed the page number zero and color number 7 should be used.

**Question 4:-** Write two classes test1 and test2.Test 1 class is using a two dimensional array suppose a[][] to store data and Test2 class is using two dimensional array b[][] to store data. Write a void main function which will sort the data and display in following manner;

| | | | | | |
|---|---|---|---|---|---|
| A[][] data is: | 1 | 23 | b[][] data is: | 1 | 24 |
| | 2 | 35 | | 2 | 46 |
| | 3 | 10 | | 3 | 4 |

4 46 4 2

Data should be displayed in following format: sorted data [position in array a] [position in array b]
It should display 0 if it does not exist in any array.
Sorted data: 2 [0][4] 4[0][3] 10 [3][0] 23[1][0] 24[0][1] 35[2][0] 46[2][4]

**Use of constructor and destructor is mandatory.**

**Question 5:-** Design a class called Message. The Message class models a simplified e-mail message, with the following member variables (with the obvious meanings):

```
string from;
string to;
string text;   //the message body
Time time_stamp;  //use the Time class.
```

Give the class the following member functions:

**a.** A constructor that takes sender & recipient, creates an empty message time-stamped with the instant of creation.

**b.** A function that appends a line of text (passed as a string) to the message body.

**c.** A Friend function that prints the entire message in the following format:

```
From: …..
To: …..
Time: …..    //use the Time class functions get_hours() etc.
Message text:
…….
```

**Question 6:-** Consider a C++ class named Fraction. An object of type Fraction will represent a fraction (a rational number), such as 3/7, -17/5, or 8/1. Your task is to write the definition of the class Fraction, The internal representation should consist of two long integers in "reduced form", and a method that will insure that they are in reduced form.
**num --**The numerator of the fraction, of type long. (The numerator of 3/7 is 3).
**Den --**The denominator of the fraction, or type long. (The denominator of 3/7 is 7).
**reduce() --** A method taking no arguments, and returning void. If r is a Fraction, then r.reduce() modifies r.num and r.den, if necessary, so that the fraction is in reduced form.

The public member function your Fraction class should consist of three methods for computing with fractions.
**plus() --** A method taking one argument of type Fraction, and returning Fraction.
r.plus(s) returns a Fraction representing the sum of r and s. Arithmetically, addition of fractions works like this:
$a/b + c/d = (ad+bc) / bd$
Your method must apply reduce() to the sum computed by this formula, before returning.

**For Questions 7, 8 and 9, after making necessary changes as desired, run the program and write the value of each variable/object defined in main function at every step in your notebook.**

**Question 7:-** Consider the following code:

```
#include<iostream.h>
#include<conio.h>
#include<dos.h>
void pos(int,int);
class stg
{
private:
        char *p;
public:
        stg(char *temp)
        {
                p=temp;
                disp;
        }
        void disp()
        {
                pos(10,30);
                cout<<p;
        }
};
void main()
{
        stg s1="ab cd";
        stg s2="fgbg";
        getch();
}
void pos(int row,int col)
{
        union REGS i;
        i.h.ah=0x02;
        i.h.bh=1;
        i.h.dh=row;
        i.h.dl=col;
        int86(0x10,&i,&i);
}
```

(a) Correct an error (if any)
(b) Explain the output:
    (i) On changing values of i.h.bh and i.h.ah
    (ii) On multiple execution after changing these values.

**Question 8:-** Modify the code for following output:

Output:

```
        Test1 function
        Display function
        Test constructor
        Test1 function
```

Display function
Test constructor
Name is :oops
AGE IS: 24
REACHED DESTRUCTOR

**CODE:**

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class test
{
	char e_name[20];
	int e_age;
	public:
		test()
		{
			disp();
			cout<<endl<<"test constructor";
		}
		disp()
		{
			test1();
			cout<<endl<<"display function";
		}
		test1()
		{
			cout<<endl<<"test1 function";
		}
		test(char *name,int age)
		{
		   strcpy(e_name,name);
		   e_age=age;
		}
		test1(char *name,int age)
		{
			strcpy(e_name,name);
			e_age=age;
		}
		~test()
		{
			cout<<"name is:"<<e_name<<endl;
			cout<<"age is:"<<e_age<<endl;
			cout<<endl<<"reached destructor";
		}
};
void main()
{
	clrscr();
	test t;
```

```
        test *p;
        p= new test;
        p->test1("oops",24);
        getch();
}
```

**Question 9:-** Change the following code using new memory allocation operator:

```
#include<alloc.h>
#include<iostream.h>
#include<conio.h>
#define MR 3
#define MC 4
class test
{
public:
        test()
        {
                int **test,i,j;
                test=(int**)malloc(MR*sizeof(int*));
                test[0]=(int *)malloc(MR*MC*sizeof(int));
                for(i=0;i<MR;i++)
                        test[i]=test[0]+i*MC;
                for(j=0;j<MR;j++)
                        cout<<test[i]<<endl;
        }

};
void main()
{
        test *t;
        getch();
}
```

**Question 10.** Modify all the class definitions of classes made in **Week 1 Lab Assignments** to include **Default Constructors, Parameterized constructors with default values, Copy Constructor and Destructors.**

Ques 9 – Convert the Stack Structure created in Week 1 Question 2 (a) to a C++ Class. After converting use the class Stack to convert the infix expression given by user to postfix expression. Use test cases from any data structure book to test your program.

E.g (sample input):[(a+b)/c-d*(g/h+k)]

Ques 10 – Use the same Stack class above to evaluate the postfix expression given by user. Use test cases from any data structure book to test your program.

E.g (sample input):**23 45** + **5** / **4** – **35 10** / **12** + *