

Lab 06 (Dated: 12-11-2021)

OOP – BCS

Task 1: **MyList** class is discussed in last class and shared with you with program name "Lec10_9.cpp". I have placed the same file with name MyList.cpp in "OOP Todays Lab" shared folder. You have to add parameterized constructor, where size should be greater than zero, default value is 5. Assign random values to array elements as written in non-parameterized constructor. Try to avoid **duplication** of code in both constructors for securing full marks. Further do following additions & modifications:

- modify set function, set return type "**MyList&**" return current object from function by writing "***this**"
- write get function to return corresponding element if parameter is within size, otherwise return minimum integer value by writing "**INT_MIN**" (a constant already defined)
- add function to add passed number in all elements, return current object at the end
- multiply function to multiply passed number with all elements, return current object at the end
- sort function to sort array by using any sorting technique (you know), return current object at the end

```
int main(){                                //Comments code in main and uncomment one by one to check your functions
    srand(time(0));
    MyList list1(10);
    list1.show();
    cout << "Fourth element is:" << list1.get(3) << '\n';
    list1.sort().show();
    cout << "Fourth element is:" << list1.get(3) << '\n';
    list1.set(95,1).show();
    list1.shuffle(20).show();
    list1.sort().show();
    list1.add(3).show();
    list1.multiply(2).show();    return 0;
}
```

Sample Run:

```
66 25 19 18 94 42 93 42 24 28           //Random Values, may be different in your case
Fourth element is:18
18 19 24 66 25 28 42 42 93 94           //Elements after sorting
Fourth element is:66
18 95 24 66 25 28 42 42 93 94           //95 is added at index 1
42 28 24 93 25 94 42 18 95 66           //Elements after shuffling
18 24 25 42 42 66 93 94 95 95           //Elements after sorting performed again
21 27 28 45 45 69 96 97 98 98           //Elements after addition of 3
42 54 56 90 90 138 192 194 196 196      //Elements after multiplication with 2
```

Task 2: Write following member functions for Rope class. Read comments carefully and implement setter functions called from parameterized constructors. Also overload operators called from main function:

```
class Rope{
    int ft;//any positive number greater than 0
    int inch;//0-11 only, if exceed add to ft and assign remaining to inch
    //using integer division and remainder operators, default value 0
public:
    Rope(){ ft=1;  inch=1;}
    Rope(int f, int i){ setFt(f);  setInch(i);}
    void show() const {
        cout << "Length:";
        if (ft==1 )           cout << "1 foot ";
        else if (ft>1)         cout << ft << " feet ";
        if (inch==0)           cout << '\n';
        else if (inch==1)      cout << "1 inch\n";
        else                   cout << inch << " inches\n";
    }
};
```

```

int main() {           //Comments code in main and uncomment one by one to check your functions
    Rope r1(3,4), r2(-4,6);
    Rope r3(2,-5), r4(4,19);
    cout << "Rope R1: "; r1.show();
    cout << "Rope R2: "; r2.show();
    cout << "Rope R3: "; r3.show();
    cout << "Rope R4: "; r4.show();
    cout << "Rope R2+R4: "; (r2+r4).show();
    cout << "Rope R1+=R4: "; (r3+=r4).show();
    cout << "Rope R3: "; r3.show();
    cout << "Rope R1-6 inches: "; (r1-=6).show();
    return 0;
}

```

```

Rope R1: Length:3 feet 4 inches
Rope R2: Length:1 foot 6 inches
Rope R3: Length:2 feet
Rope R4: Length:5 feet 7 inches
Rope R2+R4: Length:7 feet 1 inch
Rope R1+=R4: Length:7 feet 7 inches
Rope R3: Length:7 feet 7 inches
Rope R1-6 inches: Length:2 feet 10 inches

```

END OF LAB (Best of Luck)
