Lab 04 (Dated 9-10-2020)

1. Class Quiz Result

Incomplete QuizResult class and main function is given. Write member functions in Address class, used in main function. Write getter setter functions for all data members. Output is inside box in front of main function. Write code with all consideration of OOP concepts discussed so far like const, public, private, copy constructor.

```
class QuizResult{
       int noOfStrudents;//5-15, default is 10
       int *marks;
};
int main(){
       int n, value;
       cin >> n;
       QuizResult q1(n);
       q1.read();  //input n values
       q1.show();  QuizResult
       q2 = q1;
       cin >> n >> value;
       if (n==1)
              q1.scale (value); //increase marks of all students for quiz 1
      else
              q2.scale (value); //increase marks of all students for quiz 2
      q1.show();
      q2.show();
      return 0;
}
```

Input Format	**Output Format**
6	House No:2
7 4 5 8 5 6	7 4 5 8 5 6
1 1	8 5 6 9 6 7
	7 4 5 8 5 6

Solution:

```
class QuizResult{
    int noOfStrudents;//5-15, every class should have students between 10 to 30
    int *marks;
public:
    QuizResult(int n){
        setNoOfStudents(n);
        marks = new int [noOfStrudents];
    }
    QuizResult(const QuizResult &q){
        noOfStrudents = q.noOfStrudents;
        marks = new int [noOfStrudents];
        for (int i=0;i<noOfStrudents;i++)
           marks[i] = q.marks[i];
    }
    void setNoOfStudents(int n){
```

```
        if (n>=5 && n<=15)    noOfStrudents = n;
        else                  noOfStrudents = 10;
    }
    void read() {
        for (int i=0;i<noOfStrudents;i++)
            cin >> marks[i];
    }
    void show() const{
        for (int i=0;i<noOfStrudents;i++)
            cout << marks[i] << ' ';
        cout << '\n';
    }
    void scale(int m) {
        for (int i=0;i<noOfStrudents;i++)
            marks[i] += m;
    }
};
int main(){
    int n, value;
    cin >> n;
    QuizResult q1(n);
    q1.read();  //input n values
    q1.show();
    QuizResult q2 = q1;
    cin >> n >> value;
    if (n==1)    q1.scale (value); //increase marks of all students for quiz 1
    else        q2.scale (value); //increase marks of all students for quiz 2
    q1.show();
    q2.show();
    return 0;
}
```

2. Class Time 12 Hours

Create class Time 12 Hours with required constructor, setters, show function and increment functions for hours, minutes and seconds. Besides, hours, minutes and seconds, store "AM" or "PM". Create four set functions with four, three, two and one parameter(s). Put valid values check in setters. The valid value of hours, minutes and seconds is respectively 1 to 12, 0 to 59 and 0 to 59. The default values for hours, minutes and seconds is 1, 0 and 0 respectively. A represents AM and P represents PM. Write four constructors that is with 4, 3, 2 and 1 parameter(s). Write add functions for three numeric attributes. In addition of seconds, if value exceeds 59, add 1 to minutes and set remaining seconds appropriately. In addition of minutes, if value exceed 59, add 1 to hours and put remaining minutes appropriately. In addition of hours, take care of transition from AM to PM. See constraints carefully.

Finally, write main function carefully. You have to create four objects. For each object, there may be different number of inputs. Therefore, first of all read a number showing number of values to read. For example, in case of 4, there will be four inputs. First is A or P, second is hours, third is minutes, and fourth is seconds. If there are four parameters, read them and call set function with four parameters. Do similar in case of three, two and one parameter. Put default values in case of lesser inputs.

Next read a number representing count of increment operations. For each increment operation, there will be three inputs. First input is object number (1 to 4). Second input representing (1,2,3) increment in hours, minutes and seconds respectively. Third input is increment value. See input, constraint and output carefully.

Input Format

4 A 3 12 25
3 P 5 16
2 A 7
1 P
3
1 3 50
1 1 10
4 2 60

Constraints

For addition of minutes and seconds, input value will be less than equal to 60.
For addition of hours, input value will be less than equal 12.

First two test cases have no transition from AM to PM or PM to AM.
Next two test cases may have transition from AM to PM or PM to AM.

Output Format

3:12:25 AM
5:16:00 PM
7:00:00 AM
1:00:00 PM
3:13:15 AM
1:13:15 PM
2:00:00 PM

Solution:

```
class Time12Hours{
    char period;
    int hours, minutes, seconds;

public:
    void set(char p, int h, int m, int s){
        period = p;
        setHours(h);
        setMinutes(m);
        setSeconds(s);
    }
    void set(char p, int h, int m){
        period = p;
        setHours(h);
        setMinutes(m);
        seconds=0;
    }
    void set(char p, int h){
        period = p;
        setHours(h);
        minutes=0, seconds=0;
    }
    void set(char p){
        period = p;
        hours=1, minutes=0, seconds=0;
    }
    void setHours(int h){
        if (h>0 && h<12)    hours=h;
        else                hours=1;
    }
    void setMinutes(int m){
        if (m>=0 && m<60)    minutes=m;
        else                minutes=0;
    }
    void setSeconds(int s){
        if (s>=0 && s<60)    seconds=s;
```

```
        else                seconds=0;
    }
    void show() const{
        cout<<hours<<':';
        if (minutes<10)     cout<<'0';
        cout<<minutes<<':';
        if (seconds<10)     cout<<'0';
        cout<<seconds;
        if (period == 'A')  cout<<" AM\n";
        else                cout<<" PM\n";

    }
    void increaseHours(int v){
        if (hours+v > 12){
            if (period == 'A')
                period = 'P';
            else
                period = 'A';
            hours = (hours+v) % 12;
        }
        else
            hours += v;
    }
    void increaseMinutes(int v){
        if (minutes+v >= 60){
            increaseHours(1);
            minutes = (minutes+v) % 60;
        }
        else
            minutes += v;
    }
    void increaseSeconds(int v){
        if (seconds+v >= 60){
            increaseMinutes(1);
            seconds = (seconds+v) % 60;
        }
        else
            seconds += v;
    }
};
void performOperation(Time12Hours &t, int attributeNo, int value){
    if (attributeNo == 1)        t.increaseHours(value);
    else if (attributeNo == 2)   t.increaseMinutes(value);
    else                         t.increaseSeconds(value);
    t.show();
}
int main() {
    char p;
    int n, h, m, s;
    Time12Hours t1, t2, t3, t4;
    //For first object
    cin >> n;
```

```
    if (n==4){
        cin >> p >> h >> m >> s;
        t1.set(p, h, m, s);
    }
    else if (n==3){
        cin >> p >> h >> m;
        t1.set(p, h, m);
    }
    else if (n==2){
        cin >> p >> h ;
        t1.set(p, h);
    }
    else{
        cin >> p;
        t1.set(p);
    }
    //For second object
    cin >> n;
    if (n==4){
        cin >> p >> h >> m >> s;
        t2.set(p, h, m, s);
    }
    else if (n==3){
        cin >> p >> h >> m;
        t2.set(p, h, m);
    }
    else if (n==2){
        cin >> p >> h ;
        t2.set(p, h);
    }
    else{
        cin >> p;
        t2.set(p);
    }
    //For third object
    cin >> n;
    if (n==4){
        cin >> p >> h >> m >> s;
        t3.set(p, h, m, s);
    }
    else if (n==3){
        cin >> p >> h >> m;
        t3.set(p, h, m);
    }
    else if (n==2){
        cin >> p >> h ;
        t3.set(p, h);
    }
    else{
        cin >> p;
        t3.set(p);
    }
```

```
    //For fourth object
    cin >> n;
    if (n==4){
        cin >> p >> h >> m >> s;
        t4.set(p, h, m, s);
    }
    else if (n==3){
        cin >> p >> h >> m;
        t4.set(p, h, m);
    }
    else if (n==2){
        cin >> p >> h ;
        t4.set(p, h);
    }
    else{
        cin >> p;
        t4.set(p);
    }
    t1.show();
    t2.show();
    t3.show();
    t4.show();
    int objectNo, attributeNo, value;
    cin >> n;
    for (int i=0 ; i < n ; i++){
        cin >> objectNo >> attributeNo >> value;
        if      (objectNo==1)    performOperation(t1, attributeNo, value);
        else if (objectNo==2)    performOperation(t2, attributeNo, value);
        else if (objectNo==3)    performOperation(t3, attributeNo, value);
        else                     performOperation(t4, attributeNo, value);
    }
    return 0;
}
```