

**Lab 05**  
**OOP – BCS**

**Task 1:** 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 public & private. Possible values for each data member is given in front of each data member. Put appropriate check in setter to stop any invalid state of object.

```
class Address{
    int houseNo;//1-8
    char floor;//a-d
    char block;//A-J
};
int main(){
    Address a1(2,'a','C');
    Address a2(4,'a','C');
    Address a3(5,'c','D');
    a1.show();
    a2.show();
    a3.show();
    a1.setFloor('d');
    a2.setFloor('c');
    cout<<"Now A2 has floor: "<<a2.getFloor()<<'\\n';
    cout<<"A3 has block:" <<a3.getBlock()<<'\\n';
    cout<<"A1 has house no: "<<a1.getHouseNo()<<'\\n';
    return 0;
}
```

```
House No: 2
Floor: a
Block: C
-----
House No: 4
Floor: a
Block: C
-----
House No: 5
Floor: c
Block: D
-----
Now A2 has floor: c
A3 has block: D
A1 has house no: 2
```

**Task 2:** Write class Time with data members (hours, minutes, seconds, format). Valid ranges for hours is 0 to 23, for minutes is 0 to 59 and for seconds is 0:59. In case of hours, minutes & seconds, assign 0 in case of invalid value. Assign 1 to format by default (24 hours' time). Add following member functions:

**Non-Parameterized Constructor:** Assign 0 to all data members

**Parameterized Constructor:** Inside call setter functions, assign 1 to format

**Set functions with 3 parameters:** Just call individual setter functions required in next line

**Setter functions to set hours, minutes and seconds:** Check valid range and assign default values

**Set functions for format:** check if value is 1 or 2, assign to **format** data member, otherwise assign 1

**Getter functions to get hours, minutes and seconds**

**Show function** Check, if format is 1, show 24 hours' time as stored in data members, just take care, print 0 to 9 as 00 to 09 means print hours, minutes and seconds in two digits. If format is 2, show 12 hours' time, therefore change hours accordingly, for example 14 means 2 PM and 17 means 5 PM, see code and output for further clarity:

```
Time t1(3,30,18);
t1.show();                                03:30:18
t1.setSeconds (70);
t1.show();                                03:30:00
Time t2;
t2.show();                                00:00:00
t2.set(19,5,8);
t2.show();                                19:05:08
t2.setFormat(2);
t2.show();                                07:05:08 PM
t2.setHours(11); //By default 0 to 11 is AM, whereas 12 to 23 is PM
t2.show();                                11:05:08 AM
```

\*\*\*\*\*

END OF LAB (Best of Luck)

\*\*\*\*\*