Lab Assignment-3

3. Programming Excerices on C++ : Classes & Objects

3.1 Write a C++ program by creating an 'Employee' class having the following functions and print the final salary.

1. **'getInfo()'** which takes the salary, number of hours of work per day of employee as parameters

2. **'AddSal()'** which adds Rs.10 to the salary of the employee if it is less than Rs. 500.

3. **'AddWork()'** which adds Rs. 5 to the salary of the employee if the number of hours of work per day is more than 6 hours.

3.2 Modify the class and the program of Problem 3.1 for handling 10 managers as objects of Employee class by using the concept of arrays of objects.

3.3 Write a program to perform the addition of time in the hours and minutes format by using the concept of objects as function arguments.
Sample Run:
T1 = 2 hours and 45 minutes
T2 = 3 hours and 30 minutes
T3 = 6 hours and 15 minutes
Here, T1, T2 and T3 represents the objects of class Time.

3.4 Write a self-sufficient class called 'Number' which maintains an int. It should have following methods in it to perform various operation on the int:

void setNumber (int n) ; *//sets n into int*
int getNumber( ) ; *//return current value of int*
void printNumber( ) ; *//prints the int*
bool isNegative( ) ; *//checks whether int is negative*
bool isDivisibleBy (int n) ; *//checks whether int is divisible by n*
int absoluteValue( ) ; *//returns absolute value of int*

3.5 Consider a shopping list of items for which a user can place an order with a dealer every month. The list includes details such as code_number and price of each item. Different Operations which can be performed are:

1. Adding an item to the list
2. Display all items of the list.

Write a C++ program using class and objects depicting above scenario. Here, use arrays as member variables in the class.

3.6 Create two classes **DM** and **DB** which store the value of distances. **DM** stores distances in metres and centimetres and **DB** in feet and inches. Write a program that can read values for the class objects and add one object of **DM** with another object of **DB**.

Use a friend function to carry out the addition operation. The object that stores the results may be a **DM** object or **DB** object, depending on the units in which the results are required.

The display should be in the format of feet and inches or metres and centimetres depending on the object on display.