Object-Oriented Programming – Fall 2014

(BS-IT-F13 Morning & Afternoon)

Lab # 9

**Important Instructions:**

* Create a **multi-file project** to accomplish each task.
* Indent your code properly.
* Use meaningful variable and function names. Follow the naming conventions.
* Make sure that there are no **dangling pointers** and **memory leaks** in your program.

**Task #0**

Show the implement of homework (Student class contains String and Date class objects) Implement the assignment operator for these classes.

**Task#1**

This program displays how composition is used. Three classes display the hours, minutes, day, month, year, and name pertaining to an event. Implement all the member function of the following classes

class Time

{     //Time class

public:

      Time();

      Time(int, int);

      void setTime(int, int);

      void getTime(int&, int&);

      void printTime();

      void incrementHours();

      void incrementMinutes();

private:

      int hr;

      int min;

};

class Date

{//Date class

public:

      Date();

      Date(int, int, int);

      void setDate(int, int, int);

      void getDate(int&, int&, int&);

      void printDate();

private:

      int month;

      int day;

      int year;

};

class Event

{//Event class

public:

      Event(int hours = 0, int minutes = 0, int m = 1,

            int d = 1, int y = 1900, string name = "Christmas");

      void setEventData(int hours, int minutes, int m, int d, int y, string name);

      void printEventData();

private:

      String eventName;

      Time eventTime;

      Date eventDay;

};

int main()

{//instantiate an object and set data for Christmas

      Event object;

      object.setEventData(6, 0, 12, 25, 2010, "Christmas");

      //print out the data for object

      object.printEventData();

      //instantiate the second object and set date for the fourth of July

      Event object2;

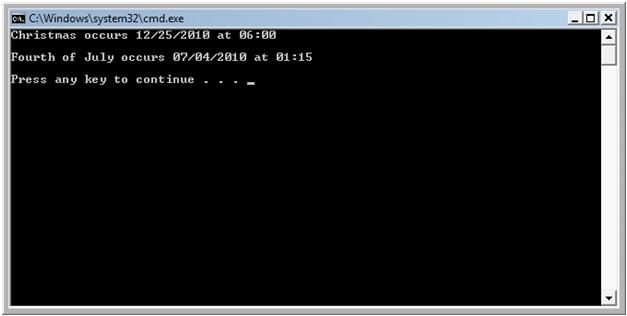
      object2.setEventData(1, 15, 7, 4, 2010, "Fourth of July");

      //print out the data for the second object

      object2.printEventData();

      return 0;

}



**Task #2**

***Note:*** *The declaration and implementation of each of the following classes should be in separate .h and .cpp files.*

1. Create a class named **Teacher**. It contains the *name* of the teacher (a c-string containing at most 40 characters) and the teacher’s *office extension number* (an integer) as member variables. Its only constructor requires the values of above two member variables as arguments.
2. Create a class named **Classroom**. It contains a *room number* (an integer) and the *capacity* (an integer) as member variables, and its only constructor requires values of both as arguments.
3. Create a class named **Course**. A Course contains a *course title* (a c-string containing at most 20 characters), a *Teacher*, and a *Classroom*. Its constructor requires the course title, the name and office extension number of the Teacher, and the room number and capacity of the Classroom.
4. Each of the above three classes should have a member function which should display all attributes of a given object.
5. Implement the **copy constructor** and **overloaded assignment operator** for each of the above three classes.
6. Write a **main()** function that creates at least three Course objects containing different values and displays them on screen.

**Task #3**

Draw a detailed **class diagram** showing the above three classes (**Teacher**, **Classroom** and **Course**) along with their member variables and functions and the relationships that exist between these classes

**Task #4**

Make the following changes in the program that you have implemented in Task #1:

1. Modify class **Teacher** so that the name of the teacher is a **char\*** instead of a statically-allocated c-string. Memory should be allocated dynamically through this pointer, depending upon the name of the teacher passed as argument into the constructor.
2. Modify the class **Course** so that the title of the course is a **char\*** instead of a statically-allocated c-string. Memory should be allocated dynamically through this pointer, depending upon the title of the course passed as argument into the constructor.
3. Implement the destructors for the classes **Teacher** and **Course**.
4. Modify the **copy constructor** and **overloaded assignment operator** for the classes **Teacher** and **Course**.
5. Write a **main()** function that *dynamically* creates at least three Course objects containing different values and displays them on screen.