

BLG252E – Object Oriented Programming
MIDTERM EXAM-1
24.10.2013
Duration: 90 minutes. Books and notes closed.

QUESTION 1) [20 points] What is the screen output, when you execute the following program ?

<pre>class A { int ID; string owner; public: A(int ID, string owner) : ID(ID), owner(owner) {cout << "CONST " << ID << " " << owner << endl; } ~A() {cout << "DEST " << ID << " " << owner << endl; } };</pre>	
<pre>void f() { cout << "f begins" << endl; A a1(1, "f normal"); static A a2(2, "f static"); A * a3 = new A(3, "f dynamic"); delete a3; cout << "f ends " << endl; }</pre>	<pre>int main() { cout << "main begins" << endl; A a5(5, "main normal"); static A a6(6, "main static"); f(); cout << "main resumes" << endl; A * a7 = new A(7, "main dynamic"); delete a7; cout << "main ends" << endl; return 0; }</pre>

QUESTION 2) [30 points] You are given the following diagram for the Clock class.

a) [20 points] Write C++ codes for the class definition and the overloaded member functions.
(All attributes are private, and all functions are public.)

Clock
hour : integer minute : integer
Clock(int, int) print() : void operator + (int) : Clock operator += (int) : Clock operator ++ () : Clock

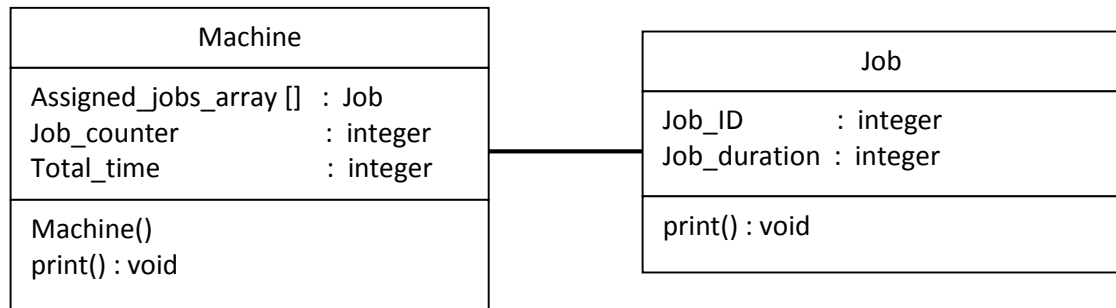
- The constructor function should initialize the hour and minute variables with the given input parameters.
- The + and += overloaded operators should take an input parameter (minute) , add it to the Clock, then return the resulting Clock object.
- The ++ overloaded operator should be a preincrement operator, which increments the Clock by 1 minute.
- The print() function should display the hour and minute in “hh:mm” format.
- Note that the valid range for hour is 0-23, for minute is 0-59. (When you add 1 minute to 23:59, the result should be 0:0)

b) [10 points] Write the main() program with the following testing scenarios:

- Declare t1 object with 23:50
- Call + operator to add 15 to t1
- Call += operator to add 20 to t1
- Print t1
- Declare t2 object
- Call + operator to assign t2 with the result of adding 10 to t1
- Call ++ operator to increment t2 by 1
- Print t2

QUESTION 3) [50 points] Suppose we have M identical machines, and N jobs (tasks) to be done. Each job takes a specified amount of time duration (hours) to complete. We would like to assign jobs to machines, so that the time at which the last job finishes is minimum. In this way, distribution of N jobs to M machines will be optimized.

a) [20 points] Using the diagram below, write C++ codes for the class definitions and the member functions. (All attributes and functions are public.)



Machine class:

- Assigned_jobs_array (max size 10) is a local array of Job objects (nesting), that will be used for storing the jobs assigned to that machine.
- Job_counter is the number of jobs assigned so far to machine.
- Total_time is the sum of job durations assigned to machine.
- Constructor initializes the Job_counter and the Total_time to zero.
- print() function displays the Assigned_jobs_array and the Total_time.

Job class:

- It contains the Job_ID number and the corresponding Job_duration.
- print() function displays the data members.

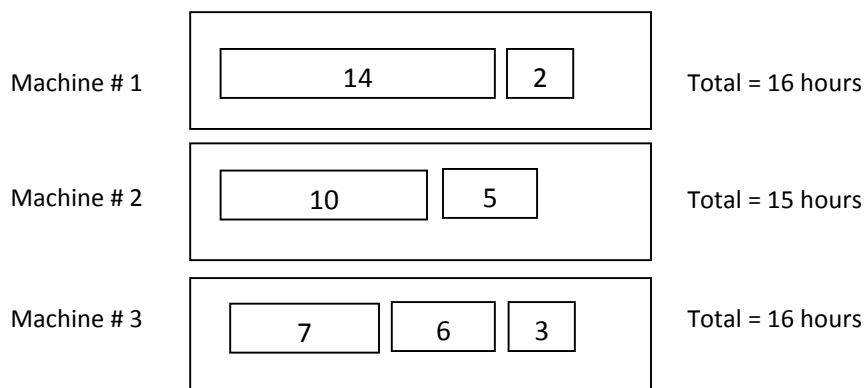
b) [30 points] Write the main() program to do the followings:

- Declare an array of Machine objects (max size 20).
- Declare an array of Job objects to store all jobs (max size 50).
- For testing purpose, set the actual number of machines as M=3, and the actual the number of jobs as N=7.
- Initialize your Jobs array with the data given below, where durations are already sorted from biggest to smallest.

Job_ID	101	102	103	104	105	106	107
Job_duration	14	10	7	6	5	3	2

- In looping statements, assign jobs to machines.

Assignment method : Each job should be scheduled (assigned) on the machine on which it will finish earliest. The following figure shows the final assignment (expected output).



- Display the results of assignments.