BLG252E – Object Oriented Programming
Midterm Exam-1
Answers

QUESTION 1) [20 points]

```
CONST 4 global normal
** main begins
CONST 5 main normal
CONST 6 main static
** f begins
CONST 1 f normal
CONST 2 f static
CONST 3 f dynamic
DEST 3 f dynamic
** f ends
DEST 1 f normal
main resumes
CONST 7 main dynamic
DEST 7 main dynamic
** main ends
Press any key to continue . . .
DEST 5 main normal
DEST 2 f static
DEST 6 main static
DEST 4 global normal

--------------------------------
Process exited with return value 0
```

QUESTION 2) [30 points]

a) [20 points]

```
#include <stdlib.h>
#include <iostream>
#include <iomanip>
using namespace std;


class Clock {
  int hour, minute;
 public:
   Clock(int h=0, int m=0) {hour=h; minute=m;}
   void print();
   Clock operator+(int m);
   Clock operator+=(int m);
   Clock operator++();
};
```

b) [10 points]

```
int main()
{
  Clock t1(23, 50);
  t1 = t1 + 15;
  t1 += 20;
  t1.print();

  Clock t2;
  t2 = t1 + 10;
  ++t2;
  t2.print();

  system("pause");
  return 0;
}
```

```
Clock Clock::operator+(int m) {
              minute += m;
              if (minute > 59)
              {
                 hour += minute / 60;
                 minute %= 60;
               }

              if (hour > 23)
                 hour %= 24;

    return *this;
}

Clock Clock::operator+=(int m) {
    *this = *this + m;  // Call overloaded operator +
    return *this;
}

Clock Clock::operator++() {
    *this = *this + 1; // Call overloaded operator +
    return *this;
}

void Clock::print() {
    cout << "Time is ";
    cout << setw(2) << setfill('0') << hour;
    cout << ":";
    cout << setw(2) << setfill('0') << minute << endl;
}
```

QUESTION 3) [50 points]

a) [20 points]

```
#include <stdlib.h>
#include <iostream>
using namespace std;

#define MMAX 20  // Maximum number of machines
#define NMAX 50  // Maximum number of jobs
#define NJ 10    // Maximum number of assigned jobs
                 //  in a machine

class Job {
  public:
  int Job_ID;
  int Job_duration;
  void print(){
    cout << Job_ID << " -- " << Job_duration << endl;
  }
};
```

b) [30 points]

```
int main()
{
  int M=3; // Number of machines for testing
  int N=7; // Number of jobs for testing

  int i,j,s;
  int ist; // Index of Smallest Total;

  Machine machines[MMAX];
  Job jobs[NMAX] = { {101, 14},{102, 10},{103, 7},
                {104, 6},{105, 5}, {106, 3}, {107, 2} };



  // Assign jobs to machines.
  // Each job is assigned on the machine
  // on which it will finish earliest.

  for (i=0; i < N; i++)
```

```
class Machine {
public:
          Job Assigned_jobs[NJ];
          int Total_time;
          int Job_counter;

          Machine() {
           Total_time=0;
           Job_counter=0;
          }

    void print(){
     cout << "Assigned jobs :\n";
     for (int i=0; i < Job_counter; i++)
        Assigned_jobs[i].print();
     cout << "Total time : " << Total_time << "\n\n";
          }
};
```

```
  {
    ist =0 ; // Index of Smallest Total;
    for (j=0; j < M; j++)
       if (machines[j].Total_time  <
machines[ist].Total_time)
         ist = j;
    int current = machines[ist].Job_counter;
    machines[ist].Assigned_jobs[current] = jobs[i];
    machines[ist].Total_time += jobs[i].Job_duration;
    machines[ist].Job_counter++;
  }

  for (j=0; j < M; j++)
  {
    cout << "Machine # " << j+1 << endl;
    machines[j].print();
  }

  system("pause");
  return 0;
}
```