1. Write a class that contains two class data members numBorn and numLiving. The value of numBorn should be equal to the number of objects of the class that have been instanced. The value of numLiving should be equal to the total number of objects in existance currently (ie, the objects that have been constructed but not yet destructed.)

2. Write a class Work whose objects represent working times (in whole minutes) and salary rates (in whole cents per minute). With this class, the following operation shall be possible:

   Work* w = new Work(25, 60); // 25 cent/min, 60 min
   w->add(65); // add 65 minutes working time
   w->printSalary(); // prints salary "31,25" (25*125 Cents)
   Work::reset(w); // reset working time to zero
   bool okay = w->subtract(60); // attempts to subtract 60 minutes
   // returns false, if not sufficient time
   // available (time remains unchanged)

   Work *v = new Work(30); // 30 cent/min, 0 min
   int r = w->compare(v); // 0 if salaries of w and v are equal,
   // 1, if w's salary is bigger, -1, else

   Work u(v); // u becomes a copy of v

3. Consider the following class definition:
   
   class TwoPoint{
   private:
       int x, y;
   public:
       void set(int x, int y);
   };

   a) Provide the definition of the method set. Write the method outside the class and do not change the prototype of the method.

   b) Write a default constructor for the class that does not have an empty parameter list.
4. What would be printed by the following program?
   a)  
      ```
      double *pt;
      double a[3]={1.2, 2.3, 3.4};
      pt=&a[1];
      pt+=1;
      cout<<*pt<<endl;
      ```
   b)  
      ```
      #include <iostream.h>
      int f(int &i)
      {
          i = 10;
          return(5 * i);
      }
      int main()
      {
          int n = 5;
          f(n);
          cout << n << "\n";
          return 0;
      }
      ```

5. How do you dynamically (a) allocate and (b) deallocate a float array X[5] by using new and delete operators?
6. Please complete the following program (by filling in lines 2, 6, 17, 24, 27, 28, 38) so that it will read data from the file test.dat and print out the data.

```cpp
#include <iostream.h>

#define MAX 10

public:
    char name[25];
    int Ex1, Ex2;
};

int main()
{
    Student st[MAX];
    int count = 0;

    if (!in)
    {
        cout << "Cannot open test.dat.\n"
        return 1;
    }

    while(!in.eof())
    {
        cout << st[count].name << " " << st[count].Ex1 << " " << st[count].Ex2 << "\n";
        count++;
        if (count >= 10) {
            cout << "Exceed MAX: " << MAX << endl;
            in.close();
            return(-1);
        }
    }

    in.close();
    return 0;
}
```

The input file test.dat contains the following data:

Eric 77 87
Scott 90 94
Mary 100 100