

Name: (as it would appear on official course roster)		section
Umail address: _____@umail.ucsb.edu		
Optional: name you wish to be called if different from name above.		
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")		

# 1

# h07

## CS16 S18

## h07: Chapter 6: File IO, Chapter 9: Dynamic memory allocation

ready?	assigned	due	points
true	Fri 05/11 09:00AM	Fri 05/18 11:59PM	100

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE, OR IF APPLICABLE, SUBMITTED ON GRADESCOPE. There is NO MAKEUP for missed assignments; in place of that, we drop the three lowest scores (if you have zeros, those are the three lowest scores.)

Read Chapter 6, Chapter 9, sections 9.1-9.2 (pages 508 - 533). The key learning goals are dynamic memory allocation and its use in dynamic arrays and linked-lists. You don't need to turn this homework in. To earn credit for this homework, complete the corresponding quiz on gauchospace AFTER you have completed the pen and pencil version of the homework. The quiz will be available one day before the due date indicated on the homework.

**PLEASE MARK YOUR HOMEWORK CLEARLY, REGARDLESS OF IF YOU WRITE IT OUT IN INK OR PENCIL! FOR BEST RESULTS, SAVE THIS PAGE AS A PDF, THEN PRINT THE PDF.**

1.(10 pts) What is the output of the following program? Using a pointer diagram show the evolution of all data objects in memory, clearly marking elements on the run-time stack and on the heap. Mark the size of all data objects. Assume the code is embedded in a correct and complete program.

```

int *p1, *p2, *p3;
p1 = new int;
p2 = new int;
p3 = p1;
*p1 = 20;
*p2 = 30;
cout<< *p1<< " " << *p2<< " " << *p3<< endl;
p1 = p2;
cout<< *p1<< " " << *p2<< " " << *p3<< endl;
*p3 = *p2;
cout<< *p1<< " " << *p2<< " " << *p3<< endl;

```

2.(10 pts) What is the output of the following program? Using a pointer diagram show the evolution of all data objects in memory, clearly marking elements on the run-time stack and on the heap. Mark the size of all data objects. Assume the code is embedded in a correct and complete program.

```

int array_size = 4, *a ;
a = new int[array_size];
int *p = a;
for(int i=0; i< array_size; i++)
    *(a+i) = 2*i;
p[0] = 10;
for(int i=0; i< array_size; i++)
    cout<<a[i]<<" ";
cout<<endl;

```

3.(10 pts) Write the definition of a structure type called UndergradStudents. This structure should contain student ID numbers, first and last names, major, and GPA scores for each undergraduate year.

```
struct Undergrad Student {
    int id;
    string first;
    string last;
    string major;
    double gpa[4];
};
```

4.(20 pts) Write a program that uses the definition of the structure UndergradStudents from the previous question to declare and then initialize an array of 3 objects of this structure (hint: you can do this with the same approach you define/initialize an array of any other type). You **must initialize the values in the program, not by user input**. The initial values are shown in the table below. Then write the definition of a function with the signature `void printRecords(struct UndergradStudents *records, int numrecords)`; The function should print out the values of the array of objects passed to it as shown in the sample below, along with each student's AVERAGE GPA score (calculated to a precision of 2 decimal places). You **must use a loop to print the output**. Your program should appropriately call the `printRecords()` function to print the student records and the average GPA.

ID	First name	Last Name	Major	GPA Yr1	GPA Yr2	GPA Yr3	GPA Yr4
1	Joe	Shmoe	EE	3.8	3.3	3.4	3.9
2	Macy	Chen	CS	3.9	3.9	4.0	4.0
3	Peter	Patrick	ME	3.8	3.0	2.4	1.9

See code on next page

OUTPUT:

```
These are the student records:
ID# 1, Shmoe, Joe, Major: EE, Average GPA: 3.60
ID# 2, Chen, Macy, Major: CS, Average GPA: 3.95
ID# 3, Peter, Patrick, Major: ME, Average GPA: 2.77
```

5.(10 pts) What is the output of this program?

```
1 #include <iostream>
2 using namespace std;
4 int main() {
5     char first, second;
6     cout << "Enter a word: ";
7     first = cin.get();
8     cin.sync();
9     second = cin.get();
10    cout << first << endl;
11    cout << second << endl;
12    return 0;
13 }
```

not covered in lecture won't be on exam

(a) First

(b) Second

(c) Return first 2 letter of number from the entered word

6.(6 pts) In C++, all the files are opened in \_\_\_\_\_ mode:

(a) Binary

(b) Text

(c) Can't say <-- you can choose either binary or text when you use .open()

7.(6 pts) How many streams are automatically created when executing a program:

2

h07

CS16 S18

(a) 1

(b) 2

(c) 3

standard input / standard output / standard error

```
Q4 int main() {
    UndergradStudents ug[3] =
```

```
{ { 1, "Joe", "Shmoë", "EE", {3.8, 3.3, 3.4, 3.9} }
  { 2, "Macy", "Chea", "CS", {3.9, 3.9, 4.0, 4.0} },
  { 3, "Peter", "Patrick", "ME", {3.8, 3.0, 2.4, 1.9} } };
```

```
printRecords(ug, 3);
```

```
}
```

```
void printRecords ( UndergradStudents s, int len) {
```

```
    for (int i=0; i < len; i++) {
```

```
        double avg = 0;
```

```
        for (int j=0; j < 4; j++) {
```

```
            avg += s[i].GPA[j];
```

```
        }
```

```
        cout << "ID# " << s[i].id << ", " << s[i].last << ", " <<
        << s[i].first << " Major: " << s[i].major << ", " <<
```

```
        << "Average GPA: " << avg << endl;
```

```
    }
```

8.(12 pts) Please fill the program below to make it output the expected output:

```
#include <fstream>
#include <iostream>
#include <string>
using namespace std;

int main ()
{
    string data;
    ofstream outfile;
    outfile.open("file.dat");
    cout << "Writing to the file" << endl;
    cout << "Enter class name: ";
    cin >> data;
    outfile << data << endl;
    cout << "Enter your id: ";
    cin >> data;
    cin.ignore();
    outfile << data << endl;
    outfile.close();
    ifstream infile;
    cout << "Reading from the file" << endl;
    infile.open("file.dat");
    getline(infile, data);
    cout << data << endl;
    getline(infile, data);
    cout << data << endl;
    infile.close();
    return 0;
}
```

# 3

# h07

## CS16 S18

### Output

```
Writing to the file
Enter class name: name
Enter your id: 123
Reading from the file
name
123
```

9.(6 pts) The <fstream> header provides three classes for operating file IO: ifstream, ofstream and fstream. What are these three used for respectively?

*ifstream*   *ofstream*   *fstream*

10.(10 pts) Write number 1 to 100 in a data file NOTES.TXT in C++

```
ofstream ofs;
ofs.open('NOTES.TXT');
for (int i = 1; i <= 100; i++)
    ofs << i;
ofs.close();
```