

1	2	3	4	5

Name: _____

Student ID #: _____

Section #: _____

CMSC 114

Exam #1

Spring 2004

This is a closed book, closed notes exam. No calculators or other aids are allowed. If you have a question during the exam, please raise your hand. Each question's value is written next to its number. Make sure there are six problems in the exam. First answer those questions that you find are easy! **YOU WILL LOSE POINTS IF YOU DO NOT WRITE YOUR NAME, STUDENT ID NUMBER, AND CORRECT SECTION NUMBER AT THE TOP OF THE EXAM (IN THE DESIGNATED AREA).**

All cell phones and other electronic devices must be turned off at all times and if a cell phone (or similar device) rings during the exam you will be asked to leave the exam at that time.

All answers must be written clearly and legibly - any answer that we can not read will be marked wrong. When writing code you do not need to write comments, but if you do not know the C++ syntax to do something, write a comment as to what you would do and continue to answer the question as best you can. You do not need to use long descriptive variable names, but should attempt to write code that is easy to follow and is properly indented.

1. [12 pts.] Give the complete output for the following program. If you believe there is an error, circle it and briefly describe how you would fix it and then continue to run the program as best as possible.

```
#include <iostream>
#include <iomanip>

using namespace std;

void f(float x, float r);
void f(float *y, float x = 2.0);

int main()
{
    float a[4] = {3.77, -1.4, 2.8};

    f(a[1], 1.0);
    f(a);

    cout << setfill('*');
    cout << setprecision(2);
    cout << setw(5) << a[0] << endl;
    cout << setiosflags(ios::fixed);
    cout << setw(5) << a[1] << endl;
    cout << setiosflags(ios::left);
    cout << setw(5) << a[2] << endl;
    cout << a[3] << endl;

    return 0;
}

void f(float x, float r)
{
    x += r;
}

void f(float *y, float x)
{
    *(y + 2) += x;
}
```

WRITE THE OUTPUT HERE:

This image shows a vertical rectangular sheet of white paper. It features ten evenly spaced, horizontal dashed lines running from the left edge to the right edge. These lines are typical of primary-ruled notebook paper used for teaching handwriting or basic writing skills. The lines are thin and black, set against a plain white background. There are no margins, text, or other markings on the page.

2. [16 pts.] Short Answer

- a. Give the syntax (shown in class) for explicitly promoting the integer 7 to type "float". (Hint: The answer is not "(float)7".)
- b. Suppose your program encounters an error while reading data from an istream called "inFlow". What statement will reset the flags so that you can use the stream again later?
- c. Can you subtract one pointer from another? (Yes/No).
- d. What is the TYPE of the argument in the following function call: f("apple");
- e. Can you pass an istream object as the argument for a function with an ifstream parameter? (Yes/No)
- f. Can you pass a "const" argument to a function with a "non-const" parameter? (Yes/No)
- g. Can you pass a "non-const" argument to a function with a "const" parameter? (Yes/No)
- h. If you are passing a LARGE object to a function, and the function will not need to modify the object, should you pass it by: 1. value, 2. reference, 3. const value, or 4. const reference? (Pick one)

3. [35 points] This question consists of three parts. All of the parts will use the following structure type definitions:

```
struct Person      |      struct Family
{                  |      {
    char name[40];  |          Person mom, dad, kids[10];
    int age;        |          int numKids;           // numKids is the # of entries
};                  |      };                               // used in the kids array
```

- a. Write a function called `readPeople`. The function will read data, representing people (see format below), from the stream passed in and will store them into the array passed in. You must use the prototype provided below and the third parameter (`numRead`) must be set equal to the number of people read into the array (note: initially `numRead`'s value is unknown).

The format for the input stream will look similar to this and will **contain no errors**:

```
Frank Smith, 72
Bob, 83
Billy Joe Parker, 100
Jim Baker, 17
... continues like this until the end of the file ...
```

Each line of input will contain data for one Person (name and age) and there is an unknown number of lines (and thus people) BUT you may assume the array passed in is large enough to store them all (it is NOT dynamic). Each name will always have a comma immediately after it and will never exceed 39 characters in length. There will be no spaces in front of the name, but the name itself may contain whitespace. You may assume that the stream is in a good state and there is data for at least one person waiting in it and your program should continue to read data until the end of file.

```
// Write the body of this function below:
void readPeople(istream &in, Person array[], int &numRead)
{
```

- b. Write a function called `oldestChild`. The function takes one parameter (a pointer to a `Family`) and returns a `Person` by value. Your function will go through the kids in the family and will return a copy of the oldest one. You may assume that the pointer points at a family variable and that the family has at least one child.

```
Person oldestChild(Family *p)
{
```

- c. Write a function called `selectFamilies` (see prototype below). The array `a` contains a list of `Families` (and `aSize` specifies its size). Your function will go through this list, looking for families where *all* children are under the age specified by `ageCutoff`. All such families must be copied into the array `b`, beginning with `b[0]`. Make sure that the variable `bSize` is set by your function to the number of `Families` that are copied into the array `b`. You may assume that all families in the array `a` have at least one child.

Your function must call and use the function `oldestChild` in a useful way.

```
void selectFamilies(Family a[], int aSize, Family b[], int &bSize, int ageCutoff)
{
```

4. [22 pts.] This is a 3 part question about BNF grammars.

- a. Write a BNF Grammar for the symbol $\langle XYG \rangle$. Valid sequences in this grammar consist of a sequence of x's and y's in any order HOWEVER the total number of x's and y's must be EVEN. Note that the "empty string" counts. Here are some examples of valid $\langle XYG \rangle$ sequences:

xyyy
yxxxyx
xy
xxxxxx
yxy

And some INVALID ones:

x
xyyyx
ABC

Below is the BNF defining the symbol <sandwich>.

```
<sandwich> ::= <layer><bread> | <layer><sandwich>
  <layer> ::= <bread><filler>
  <filler> ::= | <meat> | <cheese> | <filler><meat> | <filler><cheese>
    <meat> ::= chicken | turkey | salami | ham
  <cheese> ::= swiss | provolone | american | munster
  <bread> ::= rye | wheat | white
```

For each sequence of symbols below, determine whether or not the sequence is a valid instance of <sandwich>. (We are ignoring the spaces between symbols.) If the sequence is valid, provide a syntax-tree that PROVES it is valid. If the sequence is not valid, explain why (one or two sentences should be enough.)

b. rye turkey salami white

c. rye wheat swiss rye

5. [15 points] Write a program that will read a text file called “input” and copy its contents to a file called “output” one or more times. Your program will first ask the user how many copies they would like.

For example, imagine that the file ‘input’		Since 3 was entered then the file ‘output’
contained the text:		would contain the text:
abc		abc
d		d
		abc
and we run the program, as follows:		d
		abc
% a.out		d
How many copies do you want? 3		
%		and nothing more.

Notes:

- You may NOT use dynamic memory allocation for this question.
- You may assume that the files open correctly and that the input file contains at least one line.
- The length of each line in the input file is unknown.

```
#include <iostream>
#include <fstream>
using namespace std;
// write the rest of the code below:
```

This page intentionally left blank.

This page intentionally left blank.