

End of Semester Review

August 8, 2005

Continuing in Computer Science

- Courses at the University of Delaware
 - CISC 181: Introduction to Computer Science
 - C++: object-oriented programming
 - Similar structure to this course
 - Some review because you'll have a programming background
 - CISC 220: Data structures
 - C++
 - More data structures
 - Introduction to algorithms (e.g., more efficient sorting based on data structures)

August 8, 2005

Sara Sprenkle - CISC105

Final Review: Format

- Very Short Answer
- Short Answer
- Reading Code
- **Writing Code**

August 8, 2005

Sara Sprenkle - CISC105

Final Review: Topics

- File I/O
- recursive functions
- arrays (1-d and multiple dimensions)
- strings
- string functions
- command-line arguments
- pointers
- pass-by-reference
- structs
- sorting
- searching
- Plus the concepts for the midterm

August 8, 2005

Sara Sprenkle - CISC105

Data Types

- `int data[100][100][100];`
 - What are the data types for
 - `data` `int ***` (3d array of ints)
 - `data[0]` `int**` (2d array of ints)
 - `data[0][1]` `int*` (1d array of ints)
 - `data[3][2][1]` `int`

August 8, 2005

Sara Sprenkle - CISC105

Pointers

- What are they?
- How do we use them?
- What are the names of the relevant operators?

August 8, 2005

Sara Sprenkle - CISC105

Using command-line arguments

- `int main(int argc, char *argv[])`
- What do `argc` and `argv` represent?
- How do we access the 2nd word in `argv`?
 - How do we access the 3rd character in the 2nd word in `argv`?

August 8, 2005

Sara Sprenkle - CISC105

Clearing up some confusion

- NOTE: When we are working with parameters, as in the command line parameters for `main(int argc, char *argv[])`, the behavior of `argv` INSIDE the function is the same as if `argv` were a parameter listed as `char argv[][SIZE]` for some `SIZE` constant.
- However, the declarations
 - `char a[SIZE][SIZE];`
 - `char *b[SIZE];`
- are very different. The first declares a two-D array of `char`: space for `SIZE` strings `SIZE` long each. The second declares a 1-D array of pointers, with no allocated space.

August 8, 2005

Sara Sprenkle - CISC105

Clearing up some confusion

- However, the declarations
 - `char a[SIZE][SIZE];`
 - `char *b[SIZE];`
- are very different. The first declares a two-D array of `char`: space for `SIZE` strings `SIZE` long each. The second declares a 1-D array of pointers, with no allocated space.
- We can use `b` in a parameter list because someone else already made the space and is passing us the address (or we can declare our own space dynamically).

August 8, 2005

Sara Sprenkle - CISC105

Structs

- What are they good for?
- How do we use them?
- When do we use the dot operator versus the `->` operator?
- How can we create a new type from a struct?

August 8, 2005

Sara Sprenkle - CISC105

Sort

- In a function, sort three number parameters and put them in the appropriate order, so that $a < b < c$
 - Prototype of function?
 - Implementation?

August 8, 2005

Sara Sprenkle - CISC105

Sorting

- What three algorithms did we discuss for sorting?
 - Describe their important features

August 8, 2005

Sara Sprenkle - CISC105

Searching

- What searching methods did we discuss?
 - How do they work?
 - How fast are they?
 - Are there any limitations on these methods?

August 8, 2005

Sara Sprenkle - CISC105

Practice Problems

- ```
int x = 5;
int *xPtr = &x;
```
- (a) Declare a variable y and use xPtr to put the value from x into y.
  - (b) Can you change the value in x by using xPtr? If yes, write a statement that does so. If not, say why not.
  - (c) Can xPtr be used to point to a different integer? If yes, make xPtr point to y. If not, say why not.
  - (d) Write a statement to print the address of x.

August 8, 2005

Sara Sprenkle - CISC105

## Practice Problems

- Write a function strcpy that takes two char arrays of unknown size as parameters and copies the contents of the second parameter into the first. You do not need to check for erroneous input.

August 8, 2005

Sara Sprenkle - CISC105

## Practice Problems

- For the array "word" write a single complete statement that will:
  - char word[][] = {"eat", "the", "dead", "vegetable"};
  - (a) Change the 't' in "eat" to a 'p'
  - (b) Print the letter 'g' from "vegetable"
  - (c) Change the third word to "spam"
  - (d) Print out the third word

August 8, 2005

Sara Sprenkle - CISC105

## Practice Problems

```
struct Student {
 int idNum;
 char name[30];
 struct Student *next;
};
```

- Write a single line of code to do the following:
  - (a) Declare a struct Student x
  - (b) Declare a struct Student y and initialize it to id number 37 for a student named Akbar.
  - (c) Put all the values in y into x.
  - (d) Change the name in x to Jeff.
  - (e) Declare a one dimensional struct array a and give a[0] the values of y.

August 8, 2005

Sara Sprenkle - CISC105