CSCI 261 FINAL REVIEW

August 18, 2010

Classes Review

Write a Money class that uses a single private variable,

```
double dollars;
```

has a one parameter constructor,
`Money( double theDollars );`

has set and get accessor functions for setting the number of dollars,

and supports the main routine at the right.

Write the multiplication operator as a `friend` of the class, and the addition operator as a non-`friend` global function.

```
int main()
{
    Money m( -321.04 );
    Money p( 10000 );
    m = 2*m;
    m = p + m;
    cout.precision(2);
    cout.setf( ios::fixed );
    cout << m << "␣" << p << endl;
    return 0;
}
```

Dereferencing Pointers

Suppose the following variables are declared in code:

```
double x(3);
double* p( &x );
```

Using **all** of the following symbols (and **no others**), write a single C++ statement that changes the value of `x` to 4.5.

```
+ = * * * 2 ; p p p /
```

Dereferencing Pointers

Suppose the following variables are declared in code:

```
double x(3);
double* p( &x );
```

Using **all** of the following symbols (and **no others**), write a single C++ statement that changes the value of `x` to 4.5.

```
+ = * * * 2 ; p p p /
```

Now do the same for this set of symbols:

```
0 0 0 p p p [ [ [ ] ] ] 2 ; / = +
```

1d Arrays in Functions

Write the **prototype** of a `void` function that accepts a `double` array and a `const int` parameter for the number of elements to initialize.

1. Name the function `initSpace`.
2. Name the parameters `dataArray`, and `size`.

1d Arrays in Functions

Write the **prototype** of a `void` function that accepts a `double` array and a `const int` parameter for the number of elements to initialize.

1. Name the function `initSpace`.
2. Name the parameters `dataArray`, and `size`.
3. Now write the **function definition** for `initSpace`, it should set the values to:

   $$1.0, 1.0, 2.0, 3.0, 5.0, 8.0, 13.0, \ldots$$

   for as many elements as `size` permits.

`new` and `delete`

Write a C++ statement that dynamically allocates memory for an 8000 element double array.

1. Name the pointer `dataSpace`.
2. Check that the memory is valid, `exit(1)` if it is not.

`new` and `delete`

Write a C++ statement that dynamically allocates memory for an 8000 element double array.

1. Name the pointer `dataSpace`.
2. Check that the memory is valid, `exit(1)` if it is not.
3. How would you call `initSpace` from the `main` routine where `dataSpace` was allocated?

`new` and `delete`

Write a C++ statement that dynamically allocates memory for an 8000 element double array.

1. Name the pointer `dataSpace`.
2. Check that the memory is valid, `exit(1)` if it is not.
3. How would you call `initSpace` from the `main` routine where `dataSpace` was allocated?
4. Finally, write the C++ statement to free the memory allocated for the array.

finis