

CSCI 261 EXAM 2 REVIEW

August 18, 2010

Write the C++ prototype for the functions described below, in all cases, let the function name be `Function`:

- A A function without arguments that does not return a value.
- B A function without arguments that returns a `double`.
- C A function returning an integer value whose arguments are (in this order): an integer named `foo`, a character named `bar`, and a string named `foobar`. The integer argument is passed by reference, the other arguments are passed by value.

Consider the C++ source to the right.

- A What are the values of x, y, and z after line 36 has finished executing?
- B What are the values of x, y, and z after line 37 has finished executing?
- C What are the values of x, y, and z after line 38 has finished executing?
- D What are the values of x, y, and z after line 39 has finished executing?

```
11 void right_to_left( int left , int& right )
12 {
13     left = right;
14 }
15
16 void left_to_right( int left , int& right )
17 {
18     right = left;
19 }
20
21 void increment( int& left , int right )
22 {
23     left += 1;
24     right += left;
25 }
26
27 void decrement( int& left , int right )
28 {
29     right -= 1;
30     left -= right;
31 }
32
33 int main( void )
34 {
35     int x(0), y(1), z(2);
36     increment( x, y );
37     right_to_left( x, z );
38     left_to_right( x, z );
39     decrement( y, 0 );
40     return 0;
```

True or False? Arrays are always passed by value in function parameters.

True or False? Arrays are always passed by value in function parameters.

Suppose that

```
double findMaxValue( const double data[], const int elements );
```

is available to your main routine that has an array of doubles declared as:

```
double myData[SIZE];
```

How do you find the maximum value of myData?

2d Arrays

A sudoku grid has 9 rows, 9 columns, and 9 “quads”, each with a number from 1 to 9.

A **valid** sudoku grid does not have any repeated numbers in a row, column, or quad.

4	8	9	5	3	6	2	7	1
2	1	7	8	9	4	6	3	5
6	5	3	1	7	2	4	9	8
9	2	1	4	8	3	7	5	6
7	6	5	2	1	9	3	8	4
3	4	8	6	5	7	9	1	2
5	3	4	7	6	1	8	2	9
1	7	6	9	2	8	5	4	3
8	9	2	3	4	5	1	6	7

2d Arrays

Two rows of a valid sudoku grid may be swapped, resulting in another valid sudoku grid, as long as the two rows are in the first, middle, or last row of quads.

You can swap rows
4, 8, 9, 5, 3, ... and
6, 5, 3, 1, 7, ...,
but you **cannot** swap
4, 8, 9, 5, 3, ... and
1, 7, 6, 9, 2, ...

4	8	9	5	3	6	2	7	1
2	1	7	8	9	4	6	3	5
6	5	3	1	7	2	4	9	8
9	2	1	4	8	3	7	5	6
7	6	5	2	1	9	3	8	4
3	4	8	6	5	7	9	1	2
5	3	4	7	6	1	8	2	9
1	7	6	9	2	8	5	4	3
8	9	2	3	4	5	1	6	7

2d Arrays

Write a function definition for `rowSwap` that accepts:

1. A 2d array of integers named `puzzle` with 9 rows and 9 columns.
2. Row offsets `a` and `b`,

`rowSwap` should be a void function and (you guessed it), swap rows `a` and `b`.

4	8	9	5	3	6	2	7	1
2	1	7	8	9	4	6	3	5
6	5	3	1	7	2	4	9	8
9	2	1	4	8	3	7	5	6
7	6	5	2	1	9	3	8	4
3	4	8	6	5	7	9	1	2
5	3	4	7	6	1	8	2	9
1	7	6	9	2	8	5	4	3
8	9	2	3	4	5	1	6	7

Consider the declaration for Data:

```
1 #include <iostream>
2 using namespace std;
3
4 class Data {
5 public:
6     void setB( double newb );
7     void setY( int newy );
8     double getA() const;
9     double getB();
10    double getX() const;
11    Data process_data( const Data& datum ) const;
12    ostream& output( ostream& os ) const;
13    istream& input( istream& is );
14 private:
15    double setAY( double newa, int newy, int& oldy );
16    Data process_data( void ) const;
17    Data process_data( int newx, int newy );
18    double a, b, c;
19    int x, y;
20 };
```

- ▶ How many public data members does this class declare?

Consider the declaration for Data:

```
1 #include <iostream>
2 using namespace std;
3
4 class Data {
5 public:
6     void setB( double newb );
7     void setY( int newy );
8     double getA() const;
9     double getB();
10    double getX() const;
11    Data process_data( const Data& datum ) const;
12    ostream& output( ostream& os ) const;
13    istream& input( istream& is );
14 private:
15    double setAY( double newa, int newy, int& oldy );
16    Data process_data( void ) const;
17    Data process_data( int newx, int newy );
18    double a, b, c;
19    int x, y;
20 };
```

- ▶ Which member functions might change the state of the calling object?

Consider the declaration for Data:

```
1 #include <iostream>
2 using namespace std;
3
4 class Data {
5 public:
6     void setB( double newb );
7     void setY( int newy );
8     double getA() const;
9     double getB();
10    double getX() const;
11    Data process_data( const Data& datum ) const;
12    ostream& output( ostream& os ) const;
13    istream& input( istream& is );
14 private:
15    double setAY( double newa, int newy, int& oldy );
16    Data process_data( void ) const;
17    Data process_data( int newx, int newy );
18    double a, b, c;
19    int x, y;
20 };
```

- ▶ Which member functions would be considered *accessor* functions?

Consider the declaration for Data:

```
1 #include <iostream>
2 using namespace std;
3
4 class Data {
5 public:
6     void setB( double newb );
7     void setY( int newy );
8     double getA() const;
9     double getB();
10    double getX() const;
11    Data process_data( const Data& datum ) const;
12    ostream& output( ostream& os ) const;
13    istream& input( istream& is );
14 private:
15    double setAY( double newa, int newy, int& oldy );
16    Data process_data( void ) const;
17    Data process_data( int newx, int newy );
18    double a, b, c;
19    int x, y;
20 };
```

- ▶ What is wrong with the prototype on line 9?

Consider the declaration for Data:

```
1 #include <iostream>
2 using namespace std;
3
4 class Data {
5 public:
6     void setB( double newb );
7     void setY( int newy );
8     double getA() const;
9     double getB();
10    double getX() const;
11    Data process_data( const Data& datum ) const;
12    ostream& output( ostream& os ) const;
13    istream& input( istream& is );
14 private:
15    double setAY( double newa, int newy, int& oldy );
16    Data process_data( void ) const;
17    Data process_data( int newx, int newy );
18    double a, b, c;
19    int x, y;
20};
```

- ▶ Which would be valid C++ constructor prototypes?

`void Data();`

`Data(int x);`

`Data(void) const;`

`Data Data(void);`

Consider the declaration for Data:

```
1 #include <iostream>
2 using namespace std;
3
4 class Data {
5 public:
6     void setB( double newb );
7     void setY( int newy );
8     double getA() const;
9     double getB();
10    double getX() const;
11    Data process_data( const Data& datum ) const;
12    ostream& output( ostream& os ) const;
13    istream& input( istream& is );
14 private:
15    double setAY( double newa, int newy, int& oldy );
16    Data process_data( void ) const;
17    Data process_data( int newx, int newy );
18    double a, b, c;
19    int x, y;
20 };
```

- ▶ Write the prototype for Data's default constructor.

Consider the declaration for Data:

```
1 #include <iostream>
2 using namespace std;
3
4 class Data {
5 public:
6     void setB( double newb );
7     void setY( int newy );
8     double getA() const;
9     double getB();
10    double getX() const;
11    Data process_data( const Data& datum ) const;
12    ostream& output( ostream& os ) const;
13    istream& input( istream& is );
14 private:
15    double setAY( double newa, int newy, int& oldy );
16    Data process_data( void ) const;
17    Data process_data( int newx, int newy );
18    double a, b, c;
19    int x, y;
20 };
```

- ▶ Write the definition (what would be written in Data.cpp) of the default constructor.

All data members should be initialized to a value of 0.

Consider the declaration for Data:

```
1 #include <iostream>
2 using namespace std;
3
4 class Data {
5 public:
6     void setB( double newb );
7     void setY( int newy );
8     double getA() const;
9     double getB();
10    double getX() const;
11    Data process_data( const Data& datum ) const;
12    ostream& output( ostream& os ) const;
13    istream& input( istream& is );
14 private:
15    double setAY( double newa, int newy, int& oldy );
16    Data process_data( void ) const;
17    Data process_data( int newx, int newy );
18    double a, b, c;
19    int x, y;
20 };
```

- ▶ Write the implementation for the accessor function prototyped on line 6.

Consider the declaration for Data:

```
1 #include <iostream>
2 using namespace std;
3
4 class Data {
5 public:
6     void setB( double newb );
7     void setY( int newy );
8     double getA() const;
9     double getB();
10    double getX() const;
11    Data process_data( const Data& datum ) const;
12    ostream& output( ostream& os ) const;
13    istream& input( istream& is );
14 private:
15    double setAY( double newa, int newy, int& oldy );
16    Data process_data( void ) const;
17    Data process_data( int newx, int newy );
18    double a, b, c;
19    int x, y;
20 };
```

- ▶ Write the implementation for the accessor function prototyped on line 8.

Consider the declaration for Data:

```
1 #include <iostream>
2 using namespace std;
3
4 class Data {
5 public:
6     void setB( double newb );
7     void setY( int newy );
8     double getA() const;
9     double getB();
10    double getX() const;
11    Data process_data( const Data& datum ) const;
12    ostream& output( ostream& os ) const;
13    istream& input( istream& is );
14 private:
15    double setAY( double newa, int newy, int& oldy );
16    Data process_data( void ) const;
17    Data process_data( int newx, int newy );
18    double a, b, c;
19    int x, y;
20 };
```

- ▶ The member functions prototyped on lines 16 and 17 are examples of what C++ language feature?

Consider a (partial) implementation for Data:

```
3 Data Data::process_data( const Data& datum ) const
4 {
5     Data new_data;
6
7     new_data.b = (b + datum.x) / 2;
8     new_data.a = datum.c + y;
9
10    return new_data;
11 }
```

Describe what each part of line 3 means:

► The **first** Data?

Consider a (partial) implementation for Data:

```
3 Data Data::process_data( const Data& datum ) const
4 {
5     Data new_data;
6
7     new_data.b = (b + datum.x) / 2;
8     new_data.a = datum.c + y;
9
10    return new_data;
11 }
```

Describe what each part of line 3 means:

- ▶ The **first** Data?
- ▶ The **second** Data?

Consider a (partial) implementation for Data:

```
3 Data Data::process_data( const Data& datum ) const
4 {
5     Data new_data;
6
7     new_data.b = (b + datum.x) / 2;
8     new_data.a = datum.c + y;
9
10    return new_data;
11 }
```

Describe what each part of line 3 means:

- ▶ The **first** Data?
- ▶ The **second** Data?
- ▶ The **third** Data?

Consider a (partial) implementation for Data:

```
3 Data Data::process_data( const Data& datum ) const
4 {
5     Data new_data;
6
7     new_data.b = (b + datum.x) / 2;
8     new_data.a = datum.c + y;
9
10    return new_data;
11 }
```

Describe what each part of line 3 means:

- ▶ The **first** Data?
- ▶ The **second** Data?
- ▶ The **third** Data?
- ▶ The **first** const?

Consider a (partial) implementation for Data:

```
3 Data Data::process_data( const Data& datum ) const
4 {
5     Data new_data;
6
7     new_data.b = (b + datum.x) / 2;
8     new_data.a = datum.c + y;
9
10    return new_data;
11 }
```

Describe what each part of line 3 means:

- ▶ The **first** Data?
- ▶ The **second** Data?
- ▶ The **third** Data?
- ▶ The **first** const?
- ▶ The **last** const?

Consider a (partial) implementation for Data:

```
3 Data Data::process_data( const Data& datum ) const
4 {
5     Data new_data;
6
7     new_data.b = (b + datum.x) / 2;
8     new_data.a = datum.c + y;
9
10    return new_data;
11 }
```

```
1 #include <iostream>
2 #include "Data.h"
3 using namespace std;
4
5 int main()
6 {
7     Data a, b;
8
9     a.setY( 3 );
10    b.setB( 1.01 );
11
12    Data c = a.process_data( b );
13    cout << "c.a_=" << c.getA() << endl;
14    cout << "c.b_=" << c.getB() << endl;
15    return 0;
16 }
```

Assume that the previously mentioned default constructor (which initializes all data members to zero) has been implemented.

What do you expect will be printed by the application at the left?

finis