**Exercise #3 – Dynamic memory allocation, member initialization lists, static data members and functions.**

|  |  |
| --- | --- |
| Zachary Krausman | Due: 10/6/15 |

Questions

1. [10 pts] How many bytes are required to represent an object of the class Foo for each of the

|  |  |
| --- | --- |
| definitions below? |  |
| struct Foo { char a;  double b;  int c;  };  The size of this is 24 bytes. | struct Foo {  char a;  int b;  double c;  };  The size of this is 16 bytes. |

1. [5 pts] How many bytes are required to represent an object of the class Bar defined below:

struct Bar { char a;

int b;

double c;

static size\_t d;

};

The size of this is 16 bytes.

1. [5 pts ] Given the following new expression, how would you delete pa?

int \*pa = new int[10];

Delete [] pa;

1. [5 pts] Assume that the class MyClass has a default constructor. Are the following legal and if so, what would it do? If not, why not?

MyClass x = new MyClass;

Yes this is valid, it would dynamically allocate memory for x, an instance of MyClass.

MyClass \*x = new MyClass[10];

Yes this is valid, it will dynamically allocate space for a pointer to the beginning of a list of 10 classes.

1. [5 pts] In the code below, insert a default constructor that initializes data member of class Foo to 0 using a member initialization list.

class Foo {

private:

const int data;

public:

Foo::Foo(): data(5); };

1. [10 pts] Assume that the class MyObject has self-reporting versions defined for the default constructor, the constructor that takes a single integer argument, the assignment operator, and the destructor. What functions, and in what order, will be called when the function foo() is called:

void foo() {

MyObject x(1);

MyObject y = x; MyObject z(2); y = z;

MyObject \*w = new MyObject [10];

}

First, a constructor that accepts an integer will be called (MyObject x(1)). Next a copy constructor will be called to copy the values of x into y (MyObject y = x). Then the same constructor used on x will be called on z (MyObject z(2)). After that an overloaded assignment operator will be called the copy the values from z onto y (MyObject z(2)). And lastly the default constructor will be called ten times when the list of classes are created (MyObject \*w = new MyObject [10]).

1. [10 pts] There are two problems with the following code using the class Record as defined below. What are they and how would you fix them?

Class definition:

class Record

{ private:

const static int count = 1; 🡨 Should be static int count;

int ID; public:

Record() : ID(count) { count++; }

int getID() { return ID; }

}; 🡨 After this line, the line int Record::count = 1; should be inserted

Client code:

int main() {

Record m;

cout << m.ID << endl; 🡨 Should be m.getID(), because ID is private

}