CSCI 112 NAME \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

October 2, 2008 Minutes used: \_\_\_\_\_\_

1. (6 points) Give an example of an implicit type conversion. Explain your answer.

2. (6 points) What does the following code do?

assert(expression);

3. (6 points) Given the following class definition:

class Dog

{

public:

Dog(int age, int weight)

{m\_age = age; m\_weight = weight;}

void print()

{cout << "dog's age = " << m\_age << " weight = " << m\_weight <<endl;}

private:

int m\_age;

int m\_weight;

};

Write code to instantiate a Dog object without using new and then call it's print function. Now do the

same thing using new.

4. (8 points) In the following diagram, each box represents a file with the filename at the top of the box.

Add to the code in the boxes the necessary include directives (#include) so the program will compile.

Ignore the “...” they are simply a place holder for real code.

list.h

class List

{

public:

List();

void insert(int value);

private:

Node \*m\_head;

};

node.h

class Node

{

public:

Node(int v, Node \*next);

int m\_value;

Node \*m\_next;

};

list.cpp

List::List()

{

...

}

void List::insert(int value)

{

...

}

node.cpp

Node::Node(int v, Node \*next)

{

...

}

main.cpp

int main()

{

List the\_list;

the\_list.insert(42);

...

}

5. (8 points) What does the following program print? This question is very tricky, be careful! You must

explain your answers for any credit.

#include <iostream>

using namespace std;

int f(int value)

{

return value + 1;

}

void g(int &value)

{

value = value + 1;

}

void h(int \*value)

{

value = value + 1;

}

int main()

{

int i = 42;

f(i);

cout << i << endl;

g(i);

cout << i << endl;

h(&i);

cout << i << endl;

}

6. (8 points) What does each line in the following program print?

#include <iostream>

using namespace std;

int main()

{

int i = 42;

int \*ptr = &i;

cout << &i << endl;

cout << ptr << endl;

cout << &ptr << endl;

cout << \*ptr << endl;

cout << \*(&i) << endl;

}

For the following 4 questions, use the code on the last page. Tear off last page for easy reference.

7. (10 points) Write the constructor and the destructor for the List class.

List::List()

{

}

List::~List()

{

}

8. (16 points) Write the function bool List::is\_sorted() that returns true if the list is sorted (from smallest

number to largest number) and false if it is not sorted. Assume that an empty list is sorted.

bool List::is\_sorted()

{

}

F08 Midterm 1 5 of 8

9. (16 points) Write the function void List::insert\_sorted(int value) that inserts the given value into the list in

such a way that the list is ordered from smallest to largest. Insert the number into the list even if it is

already in the list. Assume this list is sorted when the function is called.

void List::insert\_sorted(int value)

{

}

10.(16 points) Write a function that removes the last element from the list. Return true if the list is not

empty, false if it is empty. Place the value of the node removed in the reference parameter value.

bool List::delete\_last(int &value)

{

}

Use the following class definitions for questions 7,8,9, & 10. You may not alter or add to this class

definitions.

You may tear this page off so it is easier to reference.

class Node

{

public:

Node (int value, Node \*next) {m\_value = value; m\_next = next;}

int m\_value;

Node \*m\_next;

};

class List

{

public:

List();

~List();

bool is\_sorted();

void insert\_sorted(int value);

bool delete\_value(int target);

private:

Node \*m\_head;

};