CSCE 2014 – Midterm Exam
Spring 2011

Student Name: ____________________________

Student UAID: ____________________________

Instructions: This is closed book exam. Students are allowed one 8.5x11 page of notes, but no calculators or other electronic devices. Please read all questions before starting the test and schedule your time accordingly.
Consider the following C++ code.

class Student
{
    public:
        Student();
        ~Student();
        int GetID();
        void SetGPA(float gpa);

    private:
        float GPA;
        int ID;
};

Student::~Student()
{
    GPA = 0;
    ID = 0;
}

Student::Student()
{
    GPA = 0;
    ID = 0;
}

int Student::GetID()
{
    return UAID;
}

void Student::SetGPA(float gpa)
{
    GPA = gpa;
}

[10 points] Which of the following calls uses correct C++ syntax? Assume it is an error to call a function with the wrong parameter data types or to ignore return values from methods. Also assume the code below is executed in the main program and the “Student john” object has been declared correctly.

Correct / Incorrect : john.ID = 654321;
Correct / Incorrect : int num = john.GetID();
Correct / Incorrect : john.SetID(123456);
Correct / Incorrect : john.SetGPA(3.14);
Correct / Incorrect : cout << john.GetGPA();
Consider the following C++ code.

```cpp
class Node
{
public:
    int Data;
    Node * Next;
};

int main()
{
    // Create nodes
    Node * a = new Node();
    Node * b = new Node();
    Node * c = new Node();
    a->Data = 42;
    c->Data = 17;
    b->Data = 25;

    // Connect nodes
    a->Next = b;
    a->Next->Next = c;
    a->Next->Next->Next = a;
    return 0;
}
```

[5 points] Draw a picture to show how the nodes a,b,c are linked together after the code above is executed (just before the return statement).

[5 points] What C++ code is needed in the main function to create a linked list with the data values in increasing numerical order? Remember to declare the head.
[15 points] Consider the three C++ functions below. In each case, show what the code would output if called from the main program with a linked list correctly initialized to contain the values 17, 25, and 42.

```cpp
void print1(Node *head)
{
    Node *tmp = head;
    while (tmp != NULL)
    {
        cout << tmp->Data << " ";
        tmp = tmp->Next;
    }
    cout << endl;
}
```

Output:

```cpp
void print2(Node *head)
{
    Node *tmp = head;
    while ((tmp = tmp->Next) != NULL)
    {
        cout << tmp->Data << " ";
        cout << endl;
    }
}
```

Output:

```cpp
void print3(Node *head)
{
    if (head != NULL)
    {
        cout << head->Data << " ";
        print3(head->Next);
    }
    else
    {
        cout << endl;
    }
}
```

Output:
Consider the following C++ code.

```cpp
int Dumb(int A, int B)
{
    cout << "dumb " << A << " " << B << endl;
    if (A > B)
        return 0;
    else if (A == B)
        return B;
    else
    {
        int C = (A+B) / 2;
        return Dumb(A, C) + 1;
    }
}
```

[5 points] Use the box method to trace the execution of this function when we call “int D = Dumb(5,1) from the main program. What value will be stored in D?

[5 points] Use the box method to trace the execution of this function when we call “int D = Dumb(2,4) from the main program. What value will be stored in D?
Consider the following C++ code.

```cpp
int Search(int Desired, int Data[], int Min, int Max)
{
    int Mid = (Min + Max) / 2;
    if (Max < Min)
        return (-1);
    else if (Data[Mid] == Desired)
        return (Mid);
    else if (Data[Mid] < Desired)
        return (Search(Desired, Data, Mid+1, Max));
    else if (Data[Mid] > Desired)
        return (Search(Desired, Data, Min, Mid-1));
}

int main()
{
    int Data[10] = {0, 2, 4, 6, 8, 10, 12, 14, 16, 18};
    cout << "search 14 " << Search(14, Data, 0, 9) << endl;
    cout << "search 3 " << Search(3, Data, 0, 9) << endl;
}
```

[5 points] Use the box method to trace the execution of the search 14 call. What will the program print?

[5 points] Use the box method to trace the execution of the search 3 call. What will the program print?
Consider the following C++ code.

```c++
void Tower(int count, char A, char B, char C)
{
    // Terminating condition
    if (count == 1)
        cout << “move disk from” << A << “to” << B;

    // Recursive case
    else
    {
        Tower(count – 1, A, C, B);
        Tower(1, A, B, C);
        Tower(count – 1, C, B, A);
    }
}
```

[5 points] Use the box method to trace the execution of this program when we call “Tower(2, ‘x’, ‘y’, ‘z’)” from the main program. How many times was the function Tower called in total?

[5 points] The number of recursive calls to Tower grows very rapidly with the number of disks to move. Based on the trace above, many calls to Tower will occur if we call “Tower(3,’x’,’y’,’z’)” from the main program? (You do not need to do a full box method trace to answer this)
Consider the following C++ code.

```cpp
class Stack
{
public:
    Stack();
    ~Stack();
    void Push(int Number);
    int Pop();
    int Top();
private:
    // Hidden
};

int main()
{
    Stack s;
    s.Push(3);
    s.Push(8);
    s.Push(6);
    int num = s.Pop();
    s.Push( s.Pop() + s.Pop() );
    s.Push(42);
    s.Push( s.Top() / 2 );
}
```

[10 points] Draw a sequence of pictures to show the contents of the stack after each of the lines of code in the main program are executed. Draw a little arrow in each picture to show where the top of the stack is located.