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. The exam has seven questions worth a total of 100 points, so schedule your time accordingly.
Consider the following C++ code.

```cpp
const int MAXSTUDENTS = 100;

class Group
{
public:
    ~Group();
    Group();
    bool FillInfo(char Filename[]);
    int GetNumStudents();
    void GetNumStudents(int &Num);
    void Print() const;
    bool ChangeGPA(char Name[], float GPA);
private:
    void SetNumStudents(int Num);
    string Names[MAXSTUDENTS];
    float GPAS[MAXSTUDENTS];
    int UAIRS[MAXSTUDENTS];
    int NumStudents;
};

Group RowingClub;
```

[10 points] Which of the following calls uses correct C++ syntax? Assume you are in the main program and variables are declared as above.

Correct / Incorrect : Print(RowingClub);

Correct / Incorrect : int MyUAID = RowingClub.UAIDS[7];

Correct / Incorrect : bool Result = RowingClub.ChangeGPA("John", 3.5);

Correct / Incorrect : RowingClub.SetNumStudents(15);

Correct / Incorrect : int NumStudents = RowingClub.GetNumStudents();
Consider the following C++ code.

class Thing
{
  public:
    ~Thing();
    Thing();
    int GetValue();
    void SetValue(int n);
  
  private:
    int Value;

};

Thing::~Thing()
{
  Value = 0;
}
Thing::Thing()
{
  Value = 0;
}
int Thing::GetValue()
{
  return Value;
}
void Thing::SetValue(int n)
{
  Value = n;
}

[10 points] Write the C++ code that declares an array of 10 Thing objects, and stores a different number in each Thing object.
Consider the following C++ code.

```cpp
int Fancy(int A, int B) {
    cout << A << " " << B << endl;
    if (B == 0)
        return A;
    else
        return Fancy(B, A % B);
}
```

[10 points] Use the box method to trace the execution of this function when we call “int C = Fancy(12, 64)” from the main program. What value will be stored in C?
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, 3, 6, 9, 12, 15, 18, 21, 24, 27};
    int A = Search(3, Data, 0, 9);
    int B = Search(42, Data, 0, 9);
}
```

[10 points] Use the box method to trace the execution of “int A = Search(3,Data,0,9)”.

[10 points] Use the box method to trace the execution of “int B = Search(42,Data,0,9)”.
Consider the following C++ code.

```cpp
class Node
{
public:
    float Score;
    int Count;
    Node* Next;
};
```

[10 points] Write the C++ code to create a linked list with TWO nodes that contains the following pairs of values (1.2, 34) and (5.6, 78), in THIS order. Assume you are in main, and declare all variables as needed.
Consider the following C++ code.

class Node
{
public:
    int Number;
    Node* Next;
};

int main()
{
    // First block of code
    Node *Head = new Node();
    Head->Number = 42;
    Head->Next = new Node();

    // Second block of code
    Node *Temp = Head->Next;
    Temp->Number = 55;
    Temp->Next = Head;
}

[10 points] Draw a picture to show what the linked list will look like after executing the first block of code above.

[10 points] Draw a picture to show what the linked list will look like after executing the second block of code above.
Consider the following C++ code.

```cpp
const int STACK_SIZE = 10;

class Stack
{
public:
    Stack();
    ~Stack();
    void Push(int Number);
    int Pop();
    void Top(int &Number);
    bool IsFull();
    bool IsEmpty();
    void Print();

private:
    int Data[STACK_SIZE];
    int Length;
};

int main()
{
    // First block of code
    Stack s;
    s.Push(3);
    s.Push(5);
    s.Push(11);
    int num = s.Pop();

    // Second block of code
    s.Push( s.Pop() – s.Pop() );
    s.Push(42);
    s.Push( s.Pop() / 2 );
}
```

Assume the stack is implemented using a fixed size array, and the bottom of the stack is in location 0, and the top of the stack is in location Length-1.

[10 points] Draw a picture of the stack after the first block of code is executed.

[10 points] Draw a picture of the stack after the second block of code is executed.