Instructions:

- Print your name in the space provided below.
- Answer each question in the space provided.
- If you want partial credit, justify your answers briefly and concisely, even when justification is not explicitly required.
- There are 13 questions, priced as marked. The maximum score is 100.
- When you have completed the test, sign the pledge at the bottom of this page and turn in the test.
- This is a closed-book, closed-notes examination. No calculators or other electronic devices may be used during this examination.
- You may not discuss (in any form: written, verbal or electronic) the content of this examination with any student who has not taken it. You must return this test form when you complete the examination. Failure to adhere to any of these restrictions is an Honor Code violation.

Do not start the test until instructed to do so!

Name ___________________________ printed

Pledge: On my honor, I have neither given nor received unauthorized aid on this examination.

KEY ___________________________ signed
Consider the following declarations:

\[ \text{int } * \text{aptr, bptr;} \]
\[ \text{int const } * \text{cptr;} \]
\[ \text{int anInteger;} \]

1. (5 points) Correct any of these declarations so they will all be syntactically correct. If they are all correct then say no changes are necessary.

The only syntax error is in the second line where there are two errors: the statement itself and the lack of initialization. If we wanted a pointer to a constant, we would have changed this to:

\[ \text{const int } * \text{cptr }= 0; \] or this could have been changed to:
\[ \text{int } * \text{const cptr }= 0; \] if we interpreted this as a constant pointer.

Consider the following node class of a linked list:

```cpp
class Node {
private:
    int data;
    Node *nextPtr;

public:
    Node (int);
    void setData (int);
    int getData() const;
    void setNextPt (Node *);
    const Node getNextPtr() const;
};
```

2. (5 points): write the destructor for this class

```cpp
Node::~Node (){
    Data = 0;
    Node = Null;
}
```

2. (10 points) For this same class, briefly discuss why you would or would not want to have an assignment method. (Do not code the assignment method if you believe the class should have one.)

This is not necessary since no dynamic memory is allocated in the class.
Consider the following class:

class Sentence {
private:
    string* Words;
public:
    Sentence(ifstream& In, int N);
    ~Sentence();
};

Sentence::Sentence(ifstream& In, int N) {
    if (N > 0) {
        Words = new string[N];
        // input code omitted
    } else Words = NULL;
}

Sentence::~Sentence() {
    delete [] Words;
}

3. (15 points) Write a copy constructor for this sentence class assuming a source sentence with N words.

    Sentence::Sentence (const Sentence & sentSource) {
        if (sentSource.Words != NULL) {
            Words = new string[N];
            for (int i=0; i<N; i++)
                Words[i] = sentSource.Words[i];
        } else Words = NULL;
    }

4. (5 points) Briefly explain when you would logically need or not need a copy constructor for this class.
   - Pass an object by value
   - Return an object by value
   - Initialize the object to an existing object
6. [5 points] In C++, when an object is used as an actual parameter and passed to an existing object by value, the formal parameter is: (2)

1) a copy of the actual parameter, made by the assignment operator.
2) a copy of the actual parameter, made by the copy constructor.
3) logically the same object as the actual parameter.
4) This is not allowed.
5) None of these

Consider the following specification for a medical office system:

Family Practice Associates wants a way of keeping track of their patients’ diagnoses, their patients’ family members, and the medications related to a specific diagnosis. Potential diagnoses are at least diabetes, high blood pressure, sickle cell anemia, and emphysema. Physicians in this practice need to be able to track relatives of patients with genetically based diseases and also need to track a particular medication for all patients in the case that recommended dosages are changed by the Federal Drug Administration.

[7 points each] Choosing from the following answers,

object class attribute behavior none

In terms of designing an object-oriented model of the Family Practice Associates record system determine whether each of the entities listed below is best characterized as a(n) ________ in the system, or if it is none. Justify your answers in one short sentence.

7. family class of thing of which there are many possible instantiations

8. track relatives This is a behavior since it describes the action of an object.

9. dosage This is an attribute of an implied drug class

10. sickle cell anemia – this is an object of the diagnosis class

11. physician This could be an attribute of a patient class or a controller class of the system

12. (10 points) Consider possible classes: diagnosis and medication. Is this an aggregation or an association? Justify your answer in one short sentence. Note its cardinality. (an example of cardinality is 1:1)
This is an association of cardinality 1:0..N since one diagnosis might have 0 to many medications; and the diagnosis and the medication are independent.

13. (10 Points) Consider possible classes: patient and diagnosis. Is this an aggregation or an association? Justify your answer in one short sentence. Is it a static or a dynamic relationship?
This is a dynamic aggregation. A diagnosis is part of a patient and does not exist without the patient, but it can change over time.