Please circle your section number:

010  013
011  014
012  015

Answer the multiple choice questions on a “Scantron Form”
Bubble in ONLY your name and your answers
DO NOT bubble in your id number or section

Answer the remaining questions directly on the exam paper.

General Instructions

• DO NOT WRITE YOUR NAME ON ANY PAGE EXCEPT THIS ONE!
• You have 1 hour 45 minutes. Pace yourself, and pay attention to the point values.
• Read all the directions carefully on each problem.
• Good luck.
The next several questions deal with the following segment of code. Notice that there are several items missing in this code, as indicated by comments containing the symbol @@@.
1. (4 pts) The function prototype for the Copy Constructor was omitted. Which of the following should be put where the comment says @@@ INSERT COPY CONSTRUCTOR HERE?

(a) void SetOfInteger_C(const SetOfInteger_C &orig);
(b) SetOfInteger_C(const SetOfInteger_C &orig);
(c) SetOfInteger_C(const SetOfInteger_C orig);
(d) SetOfInteger_C(const SetOfInteger_C &orig);

2. (4 pts) The function prototype for the Destructor was omitted. Which of the following should be put where the comment says @@@ INSERT DESTRUCTOR?

(a) void ~SetOfInteger_C(const SetOfInteger_C &orig);
(b) ~SetOfInteger_C() const;
(c) ~SetOfInteger_C();
(d) void destructor();
(e) void destructor() const;

3. (3 pts) The function prototype for overloaded assignment operator was omitted. Which of the following should be placed where the comment says: @@@ INSERT OVERLOADED ASSIGNMENT OPERATOR?

(a) SetOfInteger_C & operator = (const SetOfInteger_C &right) const;
(b) SetOfInteger_C & operator = ();
(c) SetOfInteger_C::operator = (const SetOfInteger_C &right);
(d) SetOfInteger_C::operator = ();
(e) SetOfInteger_C & operator = (const SetOfInteger_C &right);
The following main program is included for your reference, to help you understand the purpose of the SetOfIntegers class. Note that the overloaded stream insert operator has a bug, namely the "extra comma" at the end of the set. You’ll fix that bug later on this exam.

```cpp
// main.cc to test SetOfInteger_C class
#include "setOfInteger.h"
#include <iostream>
using namespace std;

int main(void)
{
    SetOfInteger_C evens;
    SetOfInteger_C odds;
    SetOfInteger_C emptySet;

    evens.addToSet(2);
    evens.addToSet(4);
    evens.addToSet(6);
    odds.addToSet(1);
    odds.addToSet(3);
    odds.addToSet(5);

    cout << "evens = " << evens << endl;
    cout << "odds = " << odds << endl;
    cout << "emptySet = " << emptySet << endl;

    SetOfInteger_C newSet;
    newSet = odds + 7;
    newSet = newSet + 11;

    cout << "newSet = " << newSet << endl;
    return 0;
}
```

stimpy[12:26am]> ./main
```
evens = {2,4,6,}
odds = {1,3,5,}
emptySet = {}
newSet = {1,3,5,7,11,}
stimpy[12:26am]>
```
4. (3 pts) Now consider this code excerpt, from the file `setOfIntegers.cc`.

```cpp
// SetOfInteger_C.cc
#include "setOfInteger.h"
#include <iostream>
#include <cassert>
using namespace std;

SetOfInteger_C::SetOfInteger_C()
{
    head = tail = NULL;
}

void SetOfInteger_C::addToSet(int newElement)
{
    if (isMemberOf(newElement)) // if newElement is already in the set
        return; // don't add (sets have no duplicates)

    // allocate new Node_S for element, and add to end of linked list
    Node_S *p = new Node_S;
    p->elem = newElement;
    p->next = NULL;

    if (head==NULL)
        head = tail = p;
    else
    {
        // @@@ INSERT CODE HERE
        }
}
```

Which of the following belongs at the spot labelled `// @@@ INSERT CODE HERE`?

(a) `tail->next = p;  
p = tail;`

(b) `tail = p;  
tail->next = p;`

(c) `p = tail;  
tail->next = p;`

(d) `tail->next = p;  
tail = p;`
5. (3 pts) The file `setOfIntegers.cc` continues as follows:

```cpp
SetOfInteger_C SetOfInteger_C::operator+(const int &right) // LEAVE OUT
{
    SetOfInteger_C result = (*this);
    // @@ INSERT CODE HERE
    return result;
}
```

Which of the following belongs at the spot labelled // @@ INSERT CODE HERE?

(a) `right.addToSet(result);`
(b) `this->addToSet(result);`
(c) `result.addToSet(right);`
(d) `addToSet(right);`

6. (3 pts) The file `setOfIntegers.cc` continues as follows:

```cpp
bool SetOfInteger_C::isMemberOf(int searchElem) const
{
    for (Node_S *p = head; p!=NULL; p=p->next )
    {
        if ( ) // @@ FILL IN MISSING CODE
            return true;
    }
    return false;
}
```

Which of the following belongs inside the if statement at the spot labelled: // @@ FILL IN MISSING CODE?

(a) `p.elem = searchElem`
(b) `p->elem = searchElem`
(c) `p.elem == searchElem`
(d) `p->elem == searchElem`
The file `setOfIntegers.cc` continues as follows:

```cpp
void SetOfInteger_C::print(ostream &out) const
{
    cout << "{";
    Node_S *p = head; p!=NULL; p=p->next)
    {
        cout << p->elem << ",";
    }
    cout << "}";
}

std::ostream & operator << (ostream & left, const SetOfInteger_C &right)
{
    // @@ INSERT CODE HERE
}
```

Which of the following belongs at the spot labelled: // @@ INSERT CODE HERE?

(a) ```cpp
right.print(left);
return left;
```  
(b) ```cpp
right.print(left);
return right;
```  
(c) ```cpp
left.print(right);
return left;
```  
(d) ```cpp
left.print(right);
return right;
```
8. (10 pts)

The next two questions refer back to the `SetOfInteger_C` class defined earlier in this exam, and the main program that accompanied it.

Consider the following print function, which is called by the overloaded stream insertion operator. As seen in the main program that appeared earlier on this exam, this function contains a bug: it prints an extra comma after the last item in the set.

```cpp
void SetOfInteger_C::print(ostream &out) const
{
    cout << "{";
    for (Node_S *p = head; p != NULL; p = p->next)
    {
        cout << p->elem << ",";
    }
    cout << "}";
}

std::ostream & operator << (ostream & left, const SetOfInteger_C &right)
{
    // @@ INSERT CODE HERE
}
```

Rewrite this function so that it does NOT print the extra comma.
Extra space in case you need it
9. Now consider the following member function definition. There is one line of code missing; without this line of code, the function fails to achieve its purpose.

```c
SetOfInteger_C::~SetOfInteger_C()
{
    Node_S *p = head;
    Node_S *trailp;
    while (p!=NULL)
    {
        trailp = p;
        p = p->next;
        // @@@ WHAT GOES HERE ???
    }
}
```

Do TWO things:

• (5 pts) Briefly describe the purpose of this function.

• (5 pts) Write the missing line of code (just the one line of code is all you need to write; you don’t need to rewrite the function).
10. Write the complete text of a file studentRecord.h that defines a class called StudentRecord. The class should contain integer data members studentNumber, totalCredits, totalQualityPoints. (6 pts)

It should also contain a constructor (with appropriate parameters to initialize all data members) and a member function gradePointAvg() (determine the return type of that one yourself). (10 pts)

Use the following formula

\[
\text{gradePointAverage} = \frac{\text{totalQualityPoints}}{\text{totalCredits}}
\]

Also include prototypes for get and set functions for studentNumber, totalCredits and totalQualityPoints. (12 pts)

Use \texttt{const} member functions everywhere that it is appropriate to do so. (5 pts)

Be sure to include an opening comment (OMIT your name), and also any necessary and appropriate pre-processor directives for preventing multiple definitions. (3 pts)
Even more extra space in case you need it
The following questions deal with this code excerpt below. Professor B. C. Dull wrote this code to illustrate how to read four strings into an array, and print them out, however as we will see the code has a problem:

```cpp
// p1.cc Illustrate how to read four words and print them out

#include <iostream>
using namespace std;

int main(void)
{
    char* word[4]; // Declare array of pointers to strings
    int i;
    for (i = 0; i < 4; i++) // Read in four words
    {
        cout << "Enter word number " << i << ":" ;
        cin >> word[i];
    }
    for (i = 0; i < 4; i++) // Print out the words that you read
    {
        cout << "word[" << i << "]=" << word[i] << endl ;
    }
    return 0;
}
```

Prof. Dull tried running this code with both CC and g++, however in both cases, he got a segmentation fault:

```
stimpy[10:21pm]> CC p1.cc -o p1
stimpy[10:21pm]> ./p1
Enter word number 0:This
Enter word number 1:is
Segmentation fault (core dumped)
stimpy[10:21pm]> g++ p1.cc -o p1
stimpy[10:21pm]> ./p1
Enter word number 0:This
Segmentation fault (core dumped)
```
11. (5 pts) Why did the program seg fault?
   For full credit, give an *specific* answer pertaining to *this* program. A “general description” of why programs have seg faults will earn only partial credit.

12. (5 pts) Fix the program so that it works correctly.
   You may rewrite the entire program, or just mark up the program text on the previous page. Either way is fine as long as your answer is *clear*.
13. Some folks, when they use Makefiles, have a habit of always typing `make clean` right before they type `make` or `make all`. However, you shouldn’t _always_ type `make clean` right before making your program; you should only do this _sometimes_.

- (5 pts) Explain why _always_ doing `make clean` at least partially defeats the purpose of using Makefiles in the first place.

- (6 pts) Describe at least one circumstance where it _would_ be advisable to do `make clean` before re-making your program, and why.

End of Exam. Total Points: 100