

Name_____

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.

Section 1. Multiple Choice

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 // setOfInteger.h
2 #ifndef SET_OF_INTEGER_H
3 #define SET_OF_INTEGER_H
4
5 #include <iostream>
6 #include <fstream>
7 using std::ostream;
8
9 class SetOfInteger_C
10 {
11 private:
12     struct Node_S
13     {
14         int elem;
15         Node_S *next;
16     };
17
18     Node_S *head;
19     Node_S *tail;
20
21 public:
22     SetOfInteger_C();
23     void addToSet(int newElement);
24     SetOfInteger_C operator+(const int &right);
25     bool isMemberOf(int searchElem) const;
26     void print(ostream &out) const;
27
28     // copy constructor
29     // @@@ INSERT COPY CONSTRUCTOR HERE
30
31     // destructor
32     // @@@ INSERT DESTRUCTOR HERE
33
34     // overloaded assignment operator
35     SetOfInteger_C & operator = (const SetOfInteger_C &right);
36 };
37 ostream & operator << (ostream & left, const SetOfInteger_C &right);
38 #endif
39
```

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(copy 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 insertion operator has a bug, namely the “extra comma” at the end of the set. You’ll fix that bug later on this exam.

```
1 // main.cc to test SetOfInteger_C class
2
3 #include "setOfInteger.h"
4 #include <iostream>
5
6 using namespace std;
7
8 int main(void)
9 {
10     SetOfInteger_C evens;
11     SetOfInteger_C odds;
12
13     SetOfInteger_C emptySet;
14
15     evens.addToSet(2);
16     evens.addToSet(4);
17     evens.addToSet(6);
18
19     odds.addToSet(1);
20     odds.addToSet(3);
21     odds.addToSet(5);
22
23     cout << "evens = " << evens << endl;
24     cout << "odds = " << odds << endl;
25     cout << "emptySet = " << emptySet << endl;
26
27     SetOfInteger_C newSet;
28
29     newSet = odds + 7;
30     newSet = newSet + 11;
31
32     cout << "newSet = " << newSet << endl;
33
34     return 0;
35 }
36
```

```
1 stimp[12:26am]> ./main
2 evens = {2,4,6,}
3 odds = {1,3,5,}
4 emptySet = {}
5 newSet = {1,3,5,7,11,}
6 stimp[12:26am]>
```

4. (3 pts) Now consider this code excerpt, from the file `setOfIntegers.cc`.

```
1 // SetOfInteger_C.cc
2 #include "setOfInteger.h"
3 #include <iostream>
4 #include <cassert>
5 using namespace std;
6
7 SetOfInteger_C::SetOfInteger_C()
8 {
9     head = tail = NULL;
10 }
11
12 void SetOfInteger_C::addToSet(int newElement)
13 {
14     if (isMemberOf(newElement)) // if newElement is already in the set
15         return;                // don't add (sets have no duplicates)
16
17     // allocate new Node_S for element, and add to end of linked list
18     Node_S *p = new Node_S;
19     p->elem = newElement;
20     p->next = NULL;
21
22     if (head==NULL)
23         head = tail = p;
24     else
25     {
26         // @@@ INSERT CODE HERE
27     }
28 }
```

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:

```
30 SetOfInteger_C SetOfInteger_C::operator+(const int &right) // LEAVE OUT
31 {
32     SetOfInteger_C result = (*this);
33     // @@@ INSERT CODE HERE
34     return result;
35 }
36 }
```

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:

```
38 bool SetOfInteger_C::isMemberOf(int searchElem) const
39 {
40     for (Node_S *p = head; p!=NULL; p=p->next )
41     {
42         if (
43             return true;
44     }
45     return false;
46 }
```

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`

7. (3 pts) The file `setOfIntegers.cc` continues as follows:

```
50 void SetOfInteger_C::print(ostream &out) const
51 {
52     cout << "{";
53     for (Node_S *p = head; p!=NULL; p=p->next)
54     {
55         cout << p->elem << ",";
56     }
57     cout << "}";
58 }
59
60 std::ostream & operator << (ostream & left, const SetOfInteger_C &right)
61 {
62     // @@@ INSERT CODE HERE
63 }
```

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

(a) `right.print(left);`
`return left;`

(b) `right.print(left);`
`return right;`

(c) `left.print(right);`
`return left;`

(d) `left.print(right);`
`return right;`

Section 2. Short Answer (answer on this sheet)

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.

```
50 void SetOfInteger_C::print(ostream &out) const
51 {
52     cout << "{";
53     for (Node_S *p =head; p!=NULL; p=p->next)
54     {
55         cout << p->elem << ",";
56     }
57     cout << "}";
58 }
59
60 std::ostream & operator << (ostream & left, const SetOfInteger_C &right)
61 {
62     // @@@ INSERT CODE HERE
63 }
```

Rewrite this function so that it does NOT print the extra comma.

Extra space in case you need it

10. Write the complete text of a file `studentRecord.h` that defines a class called `StudentRecord_C`. 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 `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)

Extra space in case you need it

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:

```
1 // p1.cc  Illustrate how to read four words and print them out
2
3 #include <iostream>
4 using namespace std;
5
6 int main(void)
7 {
8     char* word[4];           // Declare array of pointers to strings
9
10    int i;
11    for (i = 0; i < 4; i++)  // Read in four words
12    {
13        cout << "Enter word number " << i << ":" ;
14        cin >> word[i];
15    }
16
17    for (i = 0; i < 4; i++)  // Print out the words that you read
18    {
19        cout << "word[" << i << "]= " << word[i] << endl ;
20    }
21
22    return 0;
23 }
24
```

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

```
1 stimp[10:21pm]> CC p1.cc -o p1
2 stimp[10:21pm]> ./p1
3 Enter word number 0:This
4 Enter word number 1:is
5 Segmentation fault (core dumped)
6 stimp[10:21pm]> g++ p1.cc -o p1
7 stimp[10:21pm]> ./p1
8 Enter word number 0:This
9 Segmentation fault (core dumped)
10 stimp[10:21pm]>
```

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*.

