Com S 327
Fall 2016
Final Exam

DO NOT OPEN THIS EXAM UNTIL INSTRUCTED TO DO SO

Name: _____

ISU NetID (username): _____

Closed book and notes, no electronic devices, no headphones. Time limit 70 minutes. Partial credit may be given for partially correct solutions.

- Use correct C++ syntax for writing code.
- You are not required to write comments for your code; however, brief comments may help make your intention clear in case your code is incorrect.

If you have questions, please ask!

Question	Points	Your Score
1	30	
2	40	
3	30	
EC	3	
Total	100	

1. (30 pts; 5 ea) Give the output of the following code snippets, if any. If the code does not produce output, write *no output*. If the code produces a runtime error, write *runtime error*. If the code has a semantic error that will prevent compilation, write *semantic error*. There are no syntax errors, and you may assume that all required headers are included and that the containing file uses the standard namespace.

   (a)
```
cout << "If they take my stapler, "
        "then I'll set the building on fire.\n";
```

   (b)
```
string *s = string("\"Jump to Conclusions\"");

cout << "It was a " << *s << " mat." << endl;
```

   (c)
```
vector<string> days;

days.push_back(string("Sunday"));
days.push_back(string("Monday"));
days.push_back(string("Tuesday"));
days.push_back(string("Wednesday"));
days.push_back(string("Thursday"));
days.push_back(string("Friday"));
days.push_back(string("Saturday"));

cout << "Sounds like someone has a case of the "
     << days[4] << "s.\n";
```

The next three problems should be assumed to run in the same context. In other words, part e depends on part d, and part f depends on part e. These problems use the function `phrase()`, which follows.

```
// Read this function signature carefully!  It returns a
// reference to a pointer to constant character.
// What does that mean?
const char *&phrase() {
  static const char *name = "Michael Bolton";

  cout << "At least your name's not " << name << ".\n";

  return name;
}
```

(d) `phrase();`

(e) `phrase() = "Joanna";`

(f) `cout << (phrase() = string("Samir Nagheenanajar").c_str());`

2. (40 pts) Implement the methods specified given the following class. Assume that all methods are implemented—except for those which you are asked to implement—and work as their names imply (ask if you are unsure). You must implement the requested functionality fully within the assigned method; you may not alter the class declaration. An empty list is initialized with a null head and `tail`; otherwise, head addresses the first node in the list, and `tail` addresses the last.

```
class exam_list {
  class exam_list_node {
    public:
    const char *data;
    exam_list_node *next;
    exam_list_node *previous;
    inline exam_list_node(const char *d,
                          exam_list_node *n,
                          exam_list_node *p) :
      data(d), next(n), previous(p)
    {
      if (next) {
        next->previous = this;
      }
      if (previous) {
        previous->next = this;
      }
    }
  };
  private:
  exam_list_node *head;
  exam_list_node *tail;
  public:
  exam_list() : head(0), tail(0) {}
  ~exam_list() { clear(); }
  void clear();
  void insert_tail(const char *d);
  exam_list &operator=(const exam_list &el);
  friend ostream &operator<<(ostream &o, const exam_list &el);
};
```

(a) (15 pts) Implement the `clear()` method for `exam_list`, which empties the list, freeing all of the nodes, and leaves it in an empty state suitable for continued use.

(b) (25 pts) Implement the assignment operator for `exam_list`. A few hints:

- `*this` may be non-empty;
- You may use the `clear()` method if you need to, and you may assume it works; and
- A self assignment , e.g. `el = el`, is valid (if redundant) code that should not destroy the object.

3. (30 pts; 2 ea) Circle TRUE or FALSE in response to each of these statements about C++.

   Assume that the necessary headers are included for any function or class used. Read every word carefully; some of these are subtle.

   (a) The following line is a valid statement in C++:

   ```
   printf("Hello World!\n");
   ```

   TRUE FALSE

   (b) C++ is a superset of C.

   TRUE FALSE

   (c) C++ supports first class static dispatch.

   TRUE FALSE

   (d) `cout` is a function that you call to print to standard output.

   TRUE FALSE

   (e) `free()` and `delete` are interchangeable.

   TRUE FALSE

   (f) During its lifetime, a reference may refer to any number of variables.

   TRUE FALSE

   (g) Polymorphism depends on static typing.

   TRUE FALSE

(h) `dynamic_cast<>` provides a mechanism for runtime type checking of casts.

TRUE FALSE

(i) Templates are instantiated with a type at runtime.

TRUE FALSE

(j) Exceptions can be of any type.

TRUE FALSE

(k) The STL provides an exception class from which all STL exceptions are derived.

TRUE FALSE

(l) Name mangling is necessary for function overloading.

TRUE FALSE

(m) `extern "C"` tells the compiler to use C-style name mangling.

TRUE FALSE

(n) To use an object instance in C code, simply call its methods.

TRUE FALSE

(o) `const` is semantically equivalent in C and C++.

TRUE FALSE

Extra Credit. (3 pts) Write a haiku about this class.

For credit, your poem may not use any of the words *segmentation*, its abreviated form *seg*, *segfault*, *signal 11*, or *crash*. Kudos if you manage to make clear references to segmentation faults in 17 syllables without using any of these "illegal" words.

In case you're not familiar, a haiku is a poem in three lines, the first and third lines having five syllables, the second having seven. They're *supposed* to be profound. Here is an example:

Christmas time is near.
Amazon Lich Queens be damned.
Elves in the dungeon!

Okay, not so profound. Another:

A +3 poem!
I will wield it right away!
Sauron is a chump.

Also not profound. And a third:

70 minutes...
Wish I had a speed bonus.
Stop wasting my time.

Woah! Mind blown!