Name_____ Login_____

CS 012 – Practice Questions

Only bubble in one best answer per question.

1. Suppose that p1 and p2 are both pointers to the same dynamically-allocated integer value; y is a local integer variable. After the execution of which of the statements below is the pointer variable p1 considered a dangling pointer (not pointing to a valid memory address)?

a. `p1 = p2;`
b. `p1 = 0;`
c. `p2 = 0;`
d. `p1 = &y;`
e. `delete p2;`
f. `p2 = &y;`

2. Which of the following is a member function that is automatically defined for you, but one that you should define explicitly anyway if you are required to explicitly define the copy constructor to avoid a shallow copy?

a. the overloaded assignment (=) operator
b. the overloaded is_equal (==) operator
c. the destructor
d. the overloaded add-assign (+=) operator
e. a) and d)
f. a) and c)

3. Which of the following values is the closest approximation of the most number of **comparisons** needed to search for a value within an **unsorted** data set of size `100?` (i.e. uses linear search)

a. `1`
b. `100`
c. `200`
d. `10,000`
e. `7`
f. `700`

4. Which of the following values is the closest approximation of the most number of **comparisons** needed to search for a value within a **sorted** vector of size `100` using the binary search algorithm?

a. `1`
b. `100`
c. `200`
d. `10,000`
e. `7`
f. `700`

5. Which version of the `print` function would be invoked by the code:

```
Person* x = new Student;
x->print();
```

```
class Person
{ ...
  public:
   void print() const;
};
```

```
class Student : public Person
{ ...
  public:
   void print() const;
};
```

a. `Person::print()`
b. `Student::print()`
c. `void print();`
d. No function gets called, this causes a compile-time error

6. Which version of the `print` function would be called by the same code, if its declaration in the base class is changed to:

```
virtual void print() const;
```

Use the same set of responses as in the previous question

7. Static binding of a function call occurs at _____.

a. run-time
b. compile-time

8. What is output by the following code fragment?

```
int arr[] = {11, 3, 5, 8, 10, 4, 2};
int *ip = arr + 3;
cout << *ip << ' ' << ip[2] << endl;
```

a. `14 5`
b. `14 2`
c. `8 4`
d. `8 2`
e. `11 5`
f. `5 10`

9. The expression, `&p1[3]`, is equivalent to which of the following expressions?

a. `p1 += 3`
b. `p1 + 3`
c. `*p1 += 3`
d. `*p1 + 3`
e. `*(p1 += 3)`
f. `*(p1 + 3)`

10. Given the following makefile, which Linux commands will be executed (and in what order) if the user types "`make`" in the directory that this makefile is located?

```
#makefile for final exam
VAR=ls
A: C
      rm -rf *~
B:
      cp makefile makefile_copy
C: B
      $(VAR) -al
```

a.
```
rm -rf *~
```

b.
```
rm -rf *~
cp makefile makefile_copy
ls -al
```

c.
```
ls -al
rm -rf *~
```

d.
```
rm -rf *~
ls -al
```

e.
```
cp makefile makefile_copy
ls -al
rm -rf *~
```

f.
```
rm -rf *~
ls -al
cp makefile makefile_copy
```

11. Assuming the function merge has already been defined, given this recursive definition of the merge_sort function that implements the merge sort algorithm,

```
void merge_sort( vector<int> &v, int begin, int end )
{
  if ( begin >= end )
    return;
  int mid = (begin + end ) / 2;

  WHAT GOES HERE???
}
```

which of the following choices must be included at the end of this merge_sort function definition?

a. `merge(v, begin + 1, mid, end - 1);`
b. `return merge_sort(v, begin, end - 1);`
c.
```
if ( v[mid] < v[end] )
    merge_sort(v, begin, mid - 1);
```

```
else if ( v[mid] > v[begin] )
    merge_sort(v, mid + 1, end);
else
    merge(v, begin, mid, end);
```

d.
```
merge_sort(v, begin, mid);
merge_sort(v, mid + 1, end);
merge(v, begin, mid, end);
```

e.
```
merge(v, begin, mid, end);
merge_sort(v, begin, mid - 1);
merge_sort(v, mid + 1, end);
```

f.
```
return merge_sort(v, begin, mid);
return merge_sort(v, mid, end);
return merge_sort(v, begin, mid, end);
```

12. What is returned by the function call: `f("tweedleddeee")` given the following definition of the function `f`?

```
int f(const string &s)
{
  if (s.size() < 2) return 0;
  if (s.at(0) == s.at(1)) return 1 + f(s.substr(1));
  return f(s.substr(1));
}
```

a. 1 **c.** 4 **e.** 3
b. 2 **d.** 'e' **f.** 6

13. Dynamic binding refers to the process of _____.

a. giving the compiler specific instructions as to which function to use
b. delaying until run-time the choice of the appropriate function to use
c. using the scope resolution operator to decide which function to use
d. using the keyword new to allocate memory for the function code

14. Given a base class B and a derived class D that inherits from B, which of the following statements has/have a syntax error? (Assume B and D have default constructors.)

a. `B *b = new B;`
b. `D *d = new D;`
c. `B *d = new D;`
d. `D *b = new B;`
e. `c) and d) have syntax errors`
f. none of the above statements have syntax errors.

15. Which of the following pairs of classes would it make sense for them to have an inheritance relationship?

a. Zoo, Animal **c.** Car, Road **e.** All of the above
b. Drink, Coffee **d.** Car, Tire **f.** None of the above

16. Explain why a pop_back function for the IntList would have a running time of O(n) instead of O(1) like the pop_front function.

17. Given a class Base and a class Derived that is derived from Base, list all Base and Derived member functions that are executed when this function is called:

```
Derived * f() {
   return new Derived;
}
```

18. Given a class Base and a class Derived that is derived from Base, that no constructors or destructors are explicitly declared or implemented for either class, and that class Derived **inherits** the display function from class Base, list all Base and Derived member functions that are executed when this function is called:

```
void f(const Derived &d_param) {
   Derived d_local = d_param;
   d_local.display();
}
```

19. Given a class Base and a class Derived that is derived from Base, that no constructors or destructors are explicitly declared or implemented for either class, and that class Derived **extends** the display function from class Base, list all Base and Derived member functions that are executed when this function is called:

```
void f(const Derived &d_param) {
   Derived d_local = d_param;
   d_local.display();
}
```

20. Given a class Base and a class Derived that is derived from Base, that no constructors or destructors are explicitly declared or implemented for either class, and that class Derived **overrides (does not extend)** the display function from class Base, list all Base and Derived member functions that are executed when this function is called:

```
void f(const Derived &d_param) {
   Derived d_local = d_param;
   d_local.display();
}
```

21. Write a RECURSIVE function that returns a pointer to the smallest value in an array. Always return a pointer to the first instance of the smallest value if there are multiple instances of the smallest value. Return 0 (NULL) if the array is empty (size is 0).

```
const int * min(const int *arr, int size)
```

Name_____ Login_____

22. Write an IntList public member function that uses **recursion** to return the sum of all values in the list.
Hint: You may want to define a private helper function as well.

```
int IntList::sum() const
```

23. Write the definition of a **recursive** function named `find` that is passed a **c-string** and a **char** and returns with the pos of the char if it finds the char within the c-string or returns -1 if the char is not found. For example, if a c-string named `str` containing the value `"this is my string"` and a char named `c` containing the value `'i'` is passed in to `find`, the function should return the value 2. However, if `c` contained the value `'a'`, then `find` would return -1.

This is what the function call for the example above would look like:

```
int foundAt = find(str, c);
```

NOTE: This function takes a c-string, NOT a string.
NOTE: No points will be awarded if you include a loop construct within your solution.

24. Given the definitions of the classes Node and List below, find the error in the private member function `insert_after`. Come up with a test case that shows the error and then draw the list resulting from this test case and circle the error. Then write the code necessary to fix this error.

The member function `insert_after` passes in the address to a Node (Node pointer) currently in the list and an integer value to be inserted into the list. This function should then create a new Node containing this integer value passed in and insert it into the list after the Node whose address is passed in. If the address passed in is 0, then insert the new Node at the front of the List.

The `insert_after` function is a private helper function that will be used within the `insert_sorted` public member function. You DO NOT need to define the `insert_sorted` function.

Then, implement the destructor. This function should ensure there are no memory leaks.

Notice there are no `push_front`, `push_back` or `pop_front` functions declared or defined for this List class. You may NOT define these or any other member functions, even as private helper functions.

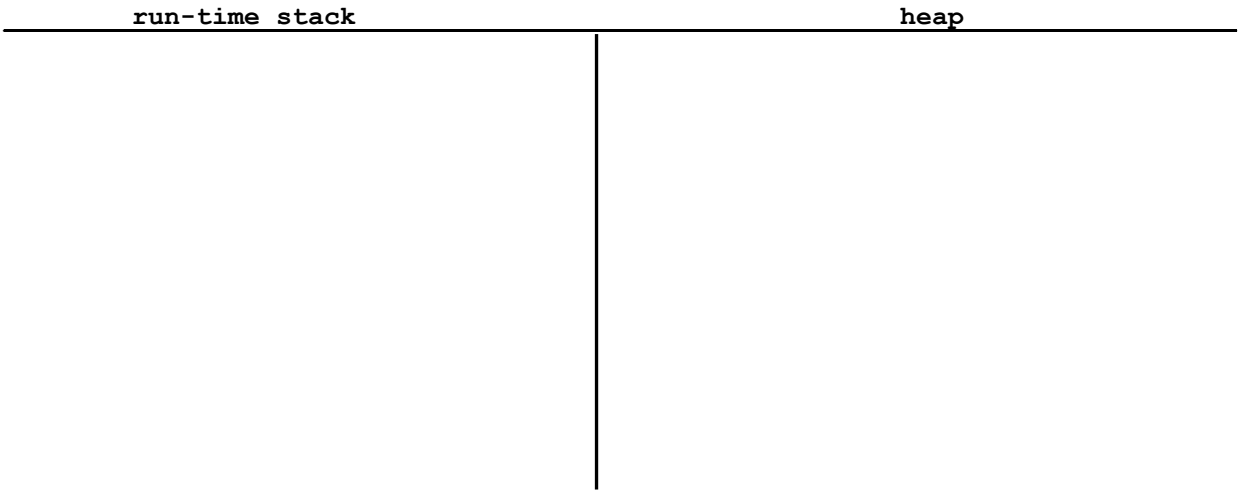
```
struct Node
{
 public:
  Node *next;
  int data;
  Node(int val)
    : next(0), data(val)
  {}
};
```

```
class List
{
 private:
  Node *head;
  Node *tail;
 public:
  List() : head(0), tail(0) {}
  ~List();
  void insert_sorted(int);
 private:
  void insert_after(Node *, int);
};
```

```
void List::insert_after(Node *prev, int val)
{
  Node *nn = new Node(val);
  if (prev == 0) //insert at front
  {
    nn->next = head;
    head = nn;
  }
  else
  {
    nn->next = prev->next;
    prev->next = nn;
  }
```

Name_____ Login_____

WHAT DOES THE MEMORY LOOK LIKE NOW?



25. Write a member function of a class MyString that overloads the + operator. This function returns a const MyString that is the concatenation of 2 MyString objects (i.e., MyString is your own version of the string class and you are overloading the + operator to work just like the string class + operator). You may use any of the functions declared below if you want. You may declare and use any private helper functions if you want. But you must also implement any private helper function you use. Something you need to take think about and take care of is making sure the cstring in cstr is large enough to store the concatenated cstrings. If needed, you may increase the capacity of cstr (i.e. the size of the cstr array) by any amount you want.

```
class MyString
{
  private:
    unsigned sz; //stores the current length of the MyString
    unsigned cap; //stores the capacity of the cstring cstr
    char *cstr; //stores a dynamically allocated cstring
  public:
    MyString(); //constructs an empty MyString (i.e., "")
    MyString(const char *); //makes a copy of the cstring
    MyString(const MyString &); //performs deep copy
    ~MyString(); //deallocates the array
    unsigned length() const; //returns the length of the string
    const char * c_str() const; //returns the cstring cstr
   //many other public member functions would be declared here

   //declare your + operator function here as a public member
   //function

  private:
   //you may declare any private helper functions here if you want
```

```
};
```

26. Overload the << operator to output your MyString object (see previous question) to either cout or an ofstream object. Be sure to show both the declaration and the implementation and specify exactly where the function is declared (inside or outside the class declaration?)

27. Write a function that sorts an array of cstrings using the selection sort algorithm.

28. Write a function that uses **stringstreams** to convert an integer to a string:

```
string int2str(int n);
```