

Student Number:

Name:

Write down answers in-between questions. Please answer using short sentences.

The back of each page can be used for practice, but DO NOT write down the answer on the back.

Be sure to write your student number and name on each page.

1. (9 pts) True / False questions:

- 1) C ++ structures can have member functions. ()
- 2) Only one constructor should exist in a class. ()
- 3) Class member functions must be specified as public. ()
- 4) If you don't set any access specifier for class members, they are considered private. ()
- 5) The elements of std :: list are stored in contiguous memory. ()
- 6) Vim editor commands are case sensitive. ()
- 7) :wg is used to quit the vim editor. ()
- 8) In the vim editor, q, w, e, r are used to move around one position from the cursor. ()
- 9) Pressing ESC in other modes to switch the vim to Normal mode. ()

2. (4 pts) Choose **ALL** incorrect implementation of the swap function that swaps the integer data x and y.

- 1)

```
void swap( int a, int b){
    int tmp = b;
    b = a;
    a = tmp;
}
```

2)

```
void swap(int* a, int* b){
    int tmp = *b;
    *b = *a;
    *a = tmp;
}
```
- 3)

```
void swap( int* const a, int* const b){
    int tmp = *b;
    *b = *a;
    *a = tmp;
}
```
- 4)

```
void swap( const int* a, const int* b){
    int tmp = *b;
    *b = *a;
    *a = tmp;
}
```

Student Number:

Name:

3. (4 pts) What is the expected result of the following code?

```
#include <iostream>
using namespace std;
namespace first_space {
    void func() {
        cout << "a";
    }
}
namespace second_space {
    void func() {
        cout << "b";
    }
}
int main () {
    using namespace first_space;
    func();
    using namespace second_space;
    func();
    return 0;
}
```

- 1) ab
 - 2) a
 - 3) b
 - 4) A compile error occurs
4. (5 pts) Select ALL the code that can be in the blank line _____ of the following code.

```
#include <iostream>
using namespace std;

class Test{
public:
    Test(int a, int b) {}
    Test(int a) {}
    Test(char c) {}
    ~Test() {}
};

int main() {
    Test _____ ;
}
```

- (a) test(1, 2);
- (b) test;
- (c) test(3);
- (d) test("a");
- (e) None of the above

Student Number:

Name:

5. (8 pts) The following program takes a string (s) and an integer (n) as command line arguments and print out the string s n times, as shown in the example below. Modify the code to work correctly.

```
1: #include <iostream>
2: #include <string>
3: #include <stdlib.h>
4: using namespace std;
5: int main(int argc, char *argv[])
6: {
7:     for(int i=0; i<argv[1]; i++)
8:         cout << argv[0] << endl;
9:
10:    return 0;
11: }
```

(Example of running the program)
\$./test abc 3
abc
abc
abc

6. (12 pts) Fill in the blanks so that the following program successfully compiles and runs without a memory leak.

```
#include <iostream>

class Vector {
public:
    int* x;
    int size;

    Vector() : x(NULL), size(0) {}
    Vector(int n) { size = n; x = _____ int[size]; }

    ~Vector() { _____ x; }

    void Resize(int n) { _____ x; x = _____ int[size = n]; }
};

int main() {
    Vector* p = _____ Vector(10);
    for (int j = 0; j < p->size; ++j) p->x[j] = j;

    Vector* q = _____ Vector[10];
    for (int i = 0; i < 10; ++i) q[i].Resize(i);

    _____ p;

    _____ q;

    return 0;
}
```

Student Number:

Name:

7. (14 pts) Fill in the blanks in the outputs of the following program.

```

#include <iostream>
using namespace std;

int main() {
    int a[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8 };
    const int num_a = sizeof(a) / sizeof(int);

    int* pa[] = { &a[0], &a[2], &a[4], &a[6], &a[8] };
    const int num_pa = sizeof(pa) / sizeof(int*);

    int* p = &a[3];
    ++p;
    *p = 9;

    const int* q = &a[0];
    q += 2;

    int*& pr1 = pa[1];
    pr1 = pa[2];

    cout << "a = {";
    for (int i = 0; i < num_a; ++i)
        cout << " " << a[i];
    cout << " }" << endl;

    cout << "=====" << endl;

    for (int i = 1; i < num_pa; ++i) --(pa[i]);
    for (int i = 1; i < num_pa; ++i) --(*pa[i]);

    cout << "a = {";
    for (int i = 0; i < num_a; ++i)
        cout << " " << a[i];
    cout << " }" << endl;

    return 0;
}

// output:
// a = { _____ (a) _____ }
// =====
// a = { _____ (b) _____ }

```

Student Number:

Name:

8. (20 pts) Write a class definition that should fit inside the box to successfully compile and run the following C++ program.

```
#include <iostream>
#include <string>
using namespace std;
```

```
void student::ShowInfo()
{
    cout << name << " is " << age << " years old, majoring " << major << "." << endl;
}

void student::SetInfo(string _name, int _age, string _major)
{
    name = _name;
    age = _age;
    major = _major;
}

void student::Study()
{
    cout << name << " is studying." << endl;
}

void student::Sleep()
{
    cout << name << " is sleeping." << endl;
}

int main()
{
    student a;
    a.SetInfo("Mary Kim", 18, "Computer Science");
    a.ShowInfo();
    a.Study();
    a.Sleep();
    return 0;
}
```

Student Number:

Name:

9. (12 pts) The following program uses a STL list container to store data. Write down the print() function that fills in the blank (a) so that the following execution results are displayed when the program is executed.

```
#include <iostream>
#include <list>
using namespace std;
void printList(const list<int>& intV) {
```

(a)

```
}

int main() {
    int n;
    cin >> n;

    list<int> intL;
    for(int i=0; i<n; ++i)
        intL.push_back(i);

    printList(intL);
    return 0;
}
```

(Example of running the program)

3↵

1 2 3

Student Number:

Name:

10. (12 pts) A palindrome is a word which reads the same backward as forward. For example, noon, level, kayak, etc. are palindromes. The `isPalin()` function below is a function that tests whether a given word is palindrome or not (returns true only if the input string `s` is palindrome). Fill in the blanks (a), (b) of the `_isPalin()` function for this to work. Also write down the output (c) – (f) when running.

```
bool _isPalin(const string& s, int begin, int end) {
    if (begin >= end-1) return true;
    return ( _____ (a) _____ ) &&
           ( _____ (b) _____ )
}

bool isPalin(const string s) {
    if (s.empty())
        return false;
    else
        return _isPalin(s, 0, s.size());
}

int main() {
    cout << "isPalin(\"abcdcdcba\")? " << isPalin("abcdcdcba") << endl;
    cout << "isPalin(\"ab\")? " << isPalin("ab") << endl;
    cout << "isPalin(\"a\")? " << isPalin("a") << endl;
    cout << "isPalin(\"\")? " << isPalin("") << endl;
}

// output:
//     isPalin("abcdcdcba")? _____ (c) _____
//     isPalin("ab")? _____ (d) _____
//     isPalin("a")? _____ (e) _____
//     isPalin("")? _____ (f) _____
```