

COMP 322 - Introduction to C++

Winter 2011

Test 1 - TA Practice Exercises

Part I: Sevan Hanssian's Practice Exercises:

1. What is the output of the following program? Show the results of intermediate steps as well.

```
# include <stdio.h>
# include <iostream>
int main()
{
    int n1 = 100;
    int n2 = 20;
    int n3 = 2;
    int n4 = -66;
    const int* a[] = {&n1, &n2, &n4};
    const int s = sizeof(a)/sizeof(*a)*2;
    std::cout << s << '\n';
    for (int i=0; i<s/3; ++i)
        printf("a[%d] = %d\n", i, *a[i+1]);
    return 0;
}
```

2. a) Find the error in the following code snippet, and briefly explain why it occurs.

```
void foo( void )
{
    char *c = "one";
    void *v = c;
    int *i = v;
}
```

b) Assuming you want the code in a) to work, how should you fix it?

3. Find any errors in the following three functions, and briefly explain why the errors occur. Also, try to explain the main differences between the three functions, and provide examples of different scenarios where one of the three functions would be useful.

```
void foo1( char* const p )
{
    p = "A";
    *p = 'A';
}
```

```
void foo2( const char* p )
{
    p = "A";
    *p = 'A';
}
```

```
void foo3( const char* const p )
{
    p = "A";
    *p = 'A';
}
```

4. Explain what happens upon compiling/running the following code snippets.

a)

```
# include <stdio.h>
```

```
void display( char* d )
{
    puts( d );
}
```

```
int main()
{
    const char* c = "two";
    display( c );
    return 0;
}
```

b)

```
int main()
{
    int x = 9;
    cout << " surprise is " << x & 1 << '\n\';
    cout << " x is "" << x ? "true\n" : "false\n";
    return 0;
}
```

c) Explain how to fix one of the code snippets in a) or b).

5. a) Write a program that uses only `iostream` methods to request data from the user, and to print out each input entered, until an end-of-file is entered. The data may be an integer number or a double. You may assume that users will not attempt to enter data of any other type. When the program encounters an end-of-file, it should print a message specifying this, and then exit.

b) Can you rewrite the program above using only `stdio` methods instead? Explain. What are the advantages of using `iostream` methods over `stdio` methods, in programs such as the one described in a)?

6. Why/when would it be useful to make a reference to a pointer? Write a short program which includes a `foo` function that *only* assigns "Hello World" to a variable (of suitable type), and a (longer) `main` that calls `foo`, such that Hello World will eventually be displayed to the screen through a reference to a pointer.

7. a) How does dynamic memory allocation take place in C++?

b) What is the output of the following program when the user enters 710?

```
# include <iostream>

int& enter(int& value)
{
    cout << "Enter an integer: ";
    cin >> value;
    return value;
}

int main()
{
    int* p = new int;
    cout << "Value = " << enter(*p) << '\n';
    delete p;
    return 0;
}
```

8. What is the output of the following program?

```
int main()
{
    int* ptr = new int(1);
    int& ref = *ptr;
    cout << ref << '\n';
    ptr = new int(2);
    cout << ref << '\n';

    int*& ref2 = ptr;
    cout << *ref2 << '\n';
    delete ref2;
    ptr = new int(2);
    cout << *ref2 << '\n';
    cout << ref << '\n';

    delete &ref;
    delete &ref2;
    delete ptr;
    return 0;
}
```

Part II: Zineng Yuan's Practice Exercises

SECTION A: Short Questions

- a. What is the main difference between *static* and *auto* variables?
- b. Could you tell the difference between **new/delete** and **malloc/free**?
- c. Anything wrong with this code?

```
T *p = new T[10];
delete p;
```
- d. What are the four functions that C++ compiler automatically provides? Write their declaration (Given the name of a class is **A**).
- e. Assuming **#define num(a,b) a - b**, could you predict the value of: **4 * num (4 + 1,2)**?

f. Read the following code, and answer the following question :

Can the test function print “hello world”? If not, explain it.

```
char *GetMemory(void)
{
    char p[ ] = "hello world";
    return p;
}
void Test(void)
{
    char *str = NULL;
    str = GetMemory();
    printf(str);
}
```

g. Read the following code, and answer the following question :

Can the test function print “hello world”? If not, explain it.

```
void GetMemory(char *p)
{
    p = (char *)malloc(100);
}
void Test(void)
{
    char *str = NULL;
    GetMemory(str);
    strcpy(str, "hello world");
    printf(str);
}
```

Section B: Programming practice

Suppose we have a Cstring class, which has a data member `char* str` and a data member `int sz`. `char *str` points to a string and `int sz` represents the length(bytes) of a Cstring object. Please complete the copy constructor and make sure it can correctly initialize a new object from a known one (Hint: you can use the function `memcpy(char *dest, char *src, unsigned int count, which copies the number of count bytes from the location pointed by src directly to the memory block pointed by dest).`

```
class Cstring{
    char * str;
    int sz;
public:
    Cstring(int size){
        sz=size;
        str = new char [sz];
    }
}
```

```
Cstring(const Cstring & obj){
//ToDo implement the copy constructor here
    ...;
}
};
```

Part III: Yancheng Xiao's Practice Exercises

1. What is the output of following program?

```
#include <iostream>
using namespace std;
#define SQR(X) X*X
int main(){
    int a = 10;
    int k = 2;
    int m = 1;
    a /= SQR(k+m) / SQR(k+m);
    cout<<a;
    return 0;
}
```

2. What is the output of the following program?

```
int main(){
    unsigned int a = 6;
    int b = -20;
    (a+b>6)?cout<<(">6"):cout<<("<=6");
    return 0;
}
```

3. When is memory allocated for a function of a class?

- a) When the program is compiled.
- b) When the function is called.

4. What is the difference between class and structure?

5. What is public, protected, private?

6. What is a namespace?

7. In the following code fragment, which is the calling object for the less-than operator?

```
string s1, s2;  
if( s1 < s2 ).
```

- a) s1
- b) <
- c) s2
- d) none

8. What error is caused by the following code ?

```
class TemperatureList {  
    public:  
        TemperatureList( );  
    private:  
        double * list;  
        int size;  
};
```

```
void func(Money amount);  
[...]
```

```
int main(){  
    TemperatureList list1;  
    func(list1);  
    return 0;  
}
```

- a) double free
- b) divide by zero
- c) illegal argument
- d) no error

9. Who can access private data in a class?

- a) members of the class
- b) friends of the class
- c) everyone
- d) A and B

10. Which of the following operators cannot be overloaded?

a) []

b) ==

c) .

d) =

Exercises Part I. 1 - 8 were inspired by the following textbook:

Learning C++: A Hands-On Approach

Eric Nagler, 1997