Homework # 8

01286131 Object-Oriented Programming

Software Engineering Program,

Department of Computer Engineering,

School of Engineering, KMITL

By

65011277 Chanasorn Howattanakulphong

Inheritance and Dynamic Binding

1. Identifying the operations in the snippets in **1.1)-1.3)**. You must answer each of the following questions:

- What objects/variables are created in the statement?
  - What are their types and values? Count the
  - number of them.
- What operations are they created? Default
  - initialization
  - Value initialization
  - Copy initialization
  - Other types of initializations
- Are there errors in the statement? Identify and correct them.
- At which statements does the copy assignment happen?
  - Which objects are assigned? By
  - what value?

1.1) Identifying the operations in the following snippet:

```
int main()
{     int ix1 = 20;
int ix2;     int
ix3();     int ix4 =
int{};

    ix2 = ix4;
     string ss1(4, '*');
string ss2{3, '&'};     string
ss3{'h', 't', 'm'};     string
ss4 = "x." + ss3;

    string ss5;     string ss6
= "x." + "png";
     ss1 = ss2 = ss4;
}
```

1.2) Identifying the operations in the following snippet:

```
int main()
{
    double num_array[6];

    string s(3, '*');

    vector<string> vs1(5, s);
vector<string> vs2(4);     vector<string>
vs3 = vs1;     vector<string> vs4{};
     vs1 = vs2 = vs3;

    vector<double> vv1{3, 2.5};
vector<double> vv2(2);     vector<double>
vv3();

    vv2 = vv1;
}
```

1.3) Identifying the operations in the following snippet:

```
int main()
{     map<string, vector<double>> x_map;
x_map["exp"] = vector<double>{1.1, 2.2, 3.3};
     vector<double> vv1 = x_map["exp"];
vector<double> vv2 = x_map["xpr"];
     vv1 = x_map["xxpr"];
vv2 = x_map["exp"];
}
```

```
/*

1.1)  Identifying the operations in the following snippet:

What objects/variables are created in the statement?
ix1, ix2, ix3, ix4, ss1, ss2, ss3, ss4, ss5, ss6

What are their types and values?
ix1: int, value: 20
ix2: int, uninitialized
ix3: function declaration, returns int, takes no arguments
ix4: int, value: 0
ss1: string, value: "****"
ss2: string, value: "&&&"
ss3: string, value: "htm"
ss4: string, value: "x.htm"
ss5: string, empty
ss6: string, value: "x.png"

Count the number of them.
10

What operations are they created?
Default initialization, value initialization, copy initialization.

Are there errors in the statement? Identify and correct them.
The declaration of int ix3() creates a function declaration instead of an integer
variable. It should be changed to int ix3 = 0;.

At which statements does the copy assignment happen?
The copy assignment happens at the statement ss1 = ss2 = ss4;.

Which objects are assigned?
ss2 and ss4 are assigned to ss1.

By what value?
ss1 is assigned the value of ss4, which is "x.htm". ss2 is assigned the value of ss4,
which is also "x.htm".

*/


/*

1.2) Identifying the operations in the following snippet:

What objects/variables are created in the statement?
num_array, s, vs1, vs2, vs3, vs4, vv1, vv2, vv3

What are their types and values?
num_array: array of doubles, uninitialized
s: string, value: ""
vs1: vector of strings, size: 5, each element initialized to ""
vs2: vector of strings, size: 4, default-initialized to empty strings
vs3: vector of strings, copy-initialized from vs1
```

```
vs4: vector of strings, size: 0
vv1: vector of doubles, size: 2, values: {3.0, 2.5}
vv2: vector of doubles, size: 2, default-initialized to 0.0
vv3: vector of doubles, default-initialized to an empty vector

Count the number of them.
9

What operations are they created?
Default initialization, value initialization, copy initialization.

Are there errors in the statement? Identify and correct them.
There are no errors.

At which statements does the copy assignment happen?
The copy assignment happens at the statements vs1 = vs2 = vs3; and vv2 = vv1;.

Which objects are assigned?
In the first statement, vs2 and vs3 are assigned to vs1. In the second statement, vv1 is
assigned to vv2.

By what value?
In the first statement, vs1 is assigned the value of vs3, which is a copy of vs1, and vs2
is assigned the value of vs3. In the second statement, vv2 is assigned the value of vv1,
which is {3.0, 2.5}.

*/

/*

1.3) Identifying the operations in the following snippet:

What objects/variables are created in the statement?
x_map, vv1, vv2

What are their types and values?
x_map: map of strings to vectors of doubles, size: 1, key: "exp", value: {1.1, 2.2, 3.3}
vv1: vector of doubles, value: {1.1, 2.2, 3.3}
vv2: vector of doubles, uninitialized

Count the number of them.
3

What operations are they created?
Default initialization, value initialization, copy assignment.

Are there errors in the statement? Identify and correct them.
There are no errors in this statement.

At which statements does the copy assignment happen?
The copy assignment happens at the statements vv1 = x_map["exp"]; and vv2 =
x_map["exp"];.

Which objects are assigned?
```

```
In the first statement, vv1 is assigned the value of the vector associated with the key
"exp" in x_map, which is {1.1, 2.2, 3.3}. In the second statement, vv2 is assigned the
same value as vv1.

By what value?
In both statements, the value assigned is {1.1, 2.2, 3.3}. In the second statement, since
the key "exp" does not exist in x_map, a default-constructed vector is returned and
assigned to vv2, which is an empty vector.

*/
```

2. Define classes of text with different word decoration and write test programs to verify their correctness.

2.1) Define classes of decorated text, `Text_base`, `Quoted_text`, and `Crypted_text`, and provide the member function `text()` for generating the decorated string from the object.

For `Text_base`, the object will store a string for the (un)decorated string `text()` will return the original string

For `Quoted_text`, the object will store a string, a string for "opening symbol", and a string for "closing symbol" `text()` will return the string decorated by quoting the original string with "opening symbol" and "closing symbol"

For `Crypted_text`, the object will store a string for the (un)decorated string `text()` will return the decorated string by encoding the letter "a–z" and "A–Z" in the original string with 'a' ⇒ 'z', 'b' ⇒ 'y', . . . , 'z' ⇒ 'a' and 'A' ⇒ 'Z', 'B' ⇒ 'Y', . . . , 'Z' ⇒ 'A'

2.2) Modify the program from **2.1)** by using the class `Text_base` as the base class for the classes of decorated text defined in **2.1)**.

*Example use cases for **2.2)***

```
    Text_base q0("Python", "*");    auto
text = q0.text();    // "Python"

    Quoted_text q("Python", "*");
const Text_base& rq = q;
    text = q.text();          // "*Python*"
text = rq.text();         // "*Python*"

    q = Quoted_text("Python", "<em>", "</em>");
text = q.text();          // "<em>Python</em>"
text = rq.text();         // "<em>Python</em>"

    Crypted_text ct("Abc101");    const
Text_base& rct = ct;    text = ct.text();
// "Zyx101"    text = rct.text();
// "Zyx101"

    ct = Crypted_text{"PYthoN101"};    text =
ct.text();          // "KBgslM101"    text =
rct.text();         // "KBgslM101"
```

2.3) Modify the program from **2.2)** by adding the class `Text` that takes `Text_base` object.

- `Text` object should behave like simple object and not requiring the use of dereference expression for accessing the object information
- Create a test program that builds a list of `Text` and iterate through them to print each object

*Test program template for **2.3)***

```
int main()
{     std::vector<Text> words{
         <<________>>("C++", "<i>", "</i>"), // Quoted_text
         <<________>>("Zidane"),             // Crypted_text
         <<________>>("Rust", "*"),          // Quoted_text
         <<________>>("Python", "[[", "]]"), // Quoted_text
         <<________>>("Vivi")                // Text_base
     };     for (const auto& w: words)
{         std::cout << w.text() <<
std::endl;     } }
```

Example output

```
<i>C++</i>
Arwzmv
*Rust*
[[Python]]
Erer
```

```
Python
*Python*
*Python*
<em>Python</em>
<em>Python</em>
Zyx101
Zyx101
KBgslM101
KBgslM101
<i>C++</i>
Arwzmv
*Rust*
[[Python]]
Erer
```