Exercises

Operators & Parameter Passing Methods

Prof. Sai-Keung Wong

TA: xyz

Questions. What are the output at the lines?

```
void main () {
        int b = 1;
        cout << b << endl;                  // L0
        cout << b-- << endl;                // L1
        cout << ++b << endl;                // L2
        cout << b++ << endl;                // L3
        cout << --b << endl;                // L4
}
```

Answers

```
void main () {
        int b = 1;
        cout << b << endl;              // L0.     1
        cout << b-- << endl;            // L1.     1
        cout << ++b << endl;            // L2.     1
        cout << b++ << endl;            // L3.     1
        cout << --b << endl;            // L4.     1
}
```

Questions. What are the output at the lines?

```
void main () {
        int b = 1;
        cout << b << endl;                  // L0
        cout << ++b << endl;                // L1
        cout << b++ << endl;                // L2
        cout << b-- << endl;                // L3
        cout << --b << endl;                // L4
}
```

Answers

```
void main () {
        int b = 1;
        cout << b << endl;                  // L0.        1
        cout << ++b << endl;                // L1.        2
        cout << b++ << endl;                // L2.        2
        cout << b-- << endl;                // L3.        3
        cout << --b << endl;                // L4.        1
}
```

Questions. What are the output at the lines?

```
int k = 12;
int g(int a) {
        k = 10;
        a = 4;
        return k + a;
}
void main () {
        int b = 1;
        cout << g(b) << endl;       // L0
        cout << b << endl;          // L1
        cout << k << endl;          // L2
}
```

Answers

```
int k = 12;
int g(int a) {
        k = 10;
        a = 4;
        return k + a;
}
void main () {
        int b = 1;
        cout << g(b) << endl;       // L0.        14
        cout << b << endl;          // L1.        1
        cout << k << endl;          // L2.        10;
}
```

Questions. What are the output at the lines?

```
int k = 12;
int g(int &a) {
        int k = 10;
        a = 4;
        return k + a;
}
void main () {
        int b = 1;
        cout << g(b) << endl;       // L0
        cout << b << endl;          // L1
        cout << k << endl;          // L2
}
```

Answers

```
int k = 12;                          // global variable
int g(int &a) {                      // pass-by-reference
        int k = 10;                  // local variable
        a = 4;                       // the actual parameter’s value is changed to 4
        return k + a;
}
void main () {
        int b = 1;
        cout << g(b) << endl;        // L0.        14
        cout << b << endl;           // L1.        4
        cout << k << endl;           // L2.        12;
}
```

Questions. What are the errors?

```
int e;                                          // L0
int g(int &a, int b) {
        static int k;                           // L1
        (e--)++;                                // L2
        a = (b>4) : e ? (++k);                  // L3
        return k + a;
}
void main () {
        int c;                                  // L4
        cout << g(2, c) << endl;                // L5
}
```

Answers

```
int e   ;                           // L0. Global variable is not initialized
                                    // before it is used.
int g(int &a, int b) {
        static int k;               // L1. Static variable is not initialized
        (e--)++;                    // L2. syntax error. (e--) is an immediate value
        a = (b>4) : e ? (++k);      // L3. syntax error. (b>4) ? e : (++k);
        return k + a;
}
void main () {
        int c;                      // L4. c is uninitialized before it is used
        cout << g(2, 3) << endl;    // L5. Formal parameter a is pass-by-reference
                                    // 2 is not a variable. 2 is an immediate value.
}
```

Questions. What are the output at the lines?

```
int g(int *a, int &b) {
        ++b;
        *a = 4 + *a;
        return (b--) + (*a);
}
void main () {
        int c = 1; int d = 2;
        cout << g(&c, d) << endl; // L0
        cout << c << endl;        // L1
        cout << d << endl;        // L2
}
```

Answers. Step 0

```
int g(int *a, int &b) {
        ++b;
        *a = 4 + *a;
        return (b--) + (*a);
}
void main () {
        int c = 1; int d = 2;
        cout << g(&c, d) << endl; // L0.        8
        cout << c << endl;        // L1.        5
        cout << d << endl;        // L2.        2
}
```

Answers. Step 1

```
int g(int *a, int &b) {                // *a = c; b = d
        ++b;                           // ++d -> d = 3, i.e., b = 3
        *a = 4 + *a;                   // *a = 4 + 1 = 5, i.e., c = 5
        return (b--) + (*a);           // return (3+5).
                                       // b--   ->  b = 2, i.e., d = 2
}
void main () {
        int c = 1; int d = 2;
        cout << g(&c, d) << endl;      // L0.         8
        cout << c << endl;             // L1.         5
        cout << d << endl;             // L2.         2
}
```

Questions [5min]. What are the output?

```
int f(int a, int b) {
    cout << a << "\t" << b << endl;

    if (a > b) {
        a--;
        return (a + 2*b) + f(a, b+1);
    }
    return a + 2*b;
}
int main( )
{
    int x = 10;
    int y = 1;
    cout << f(x, y) << endl;
}
```

Answers

```
int f(int a, int b) {
    cout << a << "\t" << b << endl;

    if (a > b) {
        a--;
        return (a + 2*b) + f(a, b+1);
    }
    return a + 2*b;
}
int main( )
{
    int x = 10;
    int y = 1;
    cout << f(x, y) << endl;
}
```

```
10    1
9     2
8     3
7     4
6     5
5     6
82
```

```
a      b
9    1
8    2
7    3
6    4
5    5
5    6
40 + 2*21
=82
```

Questions [5min]. If a is pass-by-reference, what are the output?

```
int f(int &a, int b) {
    cout << a << "\t" << b << endl;

    if (a > b) {
        a--;
        return (a + 2*b) + f(a, b+1);
    }
    return a + 2*b;
}
int main( )
{
    int x = 10;
    int y = 1;
    cout << f(x, y) << endl;
}
```

Answers. Run the program to see the result!

```
int f(int &a, int b) {
    cout << a << "\t" << b << endl;

    if (a > b) {
        a--;
        return (a + 2*b) + f(a, b+1);
    }
    return a + 2*b;
}
int main( )
{
    int x = 10;
    int y = 1;
    cout << f(x, y) << endl;
}
```