

ปฏิบัติการที่ 3 ชนิดข้อมูล string และฟังก์ชันใน C++

บทเรียนย่อย

- การประกาศตัวแปรชนิด String
- การเรียกใช้งานฟังก์ชันจากตัวแปรชนิด String
- การสร้างฟังก์ชันใน C++
- การเรียกใช้ฟังก์ชันใน C++
- การกำหนดค่าตั้งต้นให้กับพารามิเตอร์ของฟังก์ชัน
- การทำ overload ฟังก์ชัน

วัตถุประสงค์การเรียนรู้

- เรียนรู้ตัวแปรชนิด string และฟังก์ชันที่เกี่ยวข้อง
- เรียนรู้ predefined function และ user defined function
- เรียนรู้โครงสร้างฟังก์ชันใน C++
- เรียนรู้การสร้างและการเรียกใช้งานฟังก์ชัน

ให้นิสิตสร้าง Directory ชื่อว่า Lab03 สำหรับทดลองปฏิบัติการดังต่อไปนี้

ตอนที่ 1 การประกาศตัวแปรชนิด string

ใน C++ นั้นมีชนิดข้อมูล (data type) สำหรับเก็บค่า และจัดการข้อความโดยเฉพาะคือ ชนิดข้อมูล String โดยจำเป็นต้องทำการ include ไลบรารี string ที่ header คือ

```
#include <string>
```

จากนั้นสามารถประกาศใช้งานตัวแปรข้อความได้ตามตัวอย่างดังนี้

```
string name = "Hello C++";
```

การใช้งาน string มีตัวอย่างการใช้งานดังนี้

```
string str1 = "Sunny";
```

```
string str2 = str1;
```

จากตัวอย่างด้านบน str1 และ str2 มีค่าเท่ากับ "Sunny"

```
string str3 = str1 + " Day";
```

จากตัวอย่างด้านบน str3 มีค่าเท่ากับ "Sunny Day"

```
string str1 = "Hello";
```

```
string str2 = "There";
```

```
string str3 = str1 + " " + str2;
```

หรือ

```
string str3 = str1 + ' ' + str2;
```

จากตัวอย่างด้านบน str3 มีค่าเท่ากับ "Hello There"

```
string str1 = str1 + " world";
```

จากตัวอย่างด้านบน str1 มีค่าเท่ากับ "Hello world"

1. ให้นิสิตสร้างไฟล์ ex01_stringOperations.cpp มี source code ดังนี้

```
//*****
// <STUDENT_FULLNAME>
// Example string operations
//*****
#include <iostream>
#include <string>
using namespace std;
int main() {
    string name = "William Jacob";           //Line 1
    string str1, str2, str3, str4;           //Line 2
    cout << "Line 3: Name = " << name << endl; //Line 3
    str1 = "Hello There";                    //Line 4
    cout << "Line 5: str1 = " << str1 << endl; //Line 5
    str2 = str1;                             //Line 6
    cout << "Line 7: str2 = " << str2 << endl; //Line 7
    str1 = "Sunny";                          //Line 8
    str2 = str1 + " Day";                    //Line 9
    cout << "Line 10: str2 = " << str2 << endl; //Line 10
    str1 = "Hello";                         //Line 11
    str2 = "There";                         //Line 12
    str3 = str1 + " " + str2;               //Line 13
    cout << "Line 14: str3 = " << str3 << endl; //Line 14
    str3 = str1 + ' ' + str2;               //Line 15
    cout << "Line 16: str3 = " << str3 << endl; //Line 16
    str1 = str1 + " Mickey";                //Line 17
    cout << "Line 18: str1 = " << str1 << endl; //Line 18
    str1 = "Hello there";                  //Line 19
    cout << "Line 20: str1[6] = " << str1[6] << endl; //Line 20
    str1[6] = 'T';                          //Line 21
    cout << "Line 22: str1 = " << str1 << endl; //Line 22
    //String input operations
    cout << "Line 23: Enter a string with " << "no blanks: "; //Line 23
    cin >> str1;                             //Line 24
    char ch;                                //Line 25
    cin.get(ch);                            //Read the newline character; Line 26
    cout << endl;                            //Line 27
    cout << "Line 28: The string you entered = " << str1 << endl; //Line 28
    cout << "Line 29: Enter a sentence: ";    //Line 29
    getline(cin, str2);                     //Line 30
    cout << endl;                            //Line 31
    cout << "Line 32: The sentence is: " << str2 << endl; //Line 32
    return 0;
}
```

โดย getline(cin, str2) คือ การอ่านค่าจากหน้าจอทั้งบรรทัด ไปยังตัวแปร str2

2. ให้นิสิตทำความเข้าใจ source code หลังจากนั้นทำการ compile และรันโปรแกรม

3. จากการทดลองโปรแกรม เครื่องหมายบวก (+) เมื่อใช้กับ string จะมีการทำงานคือ

.....

ตอนที่ 2 การเรียกใช้งานฟังก์ชันจากตัวแปรชนิด String

ชนิดข้อมูล string คือ ชุดของตัวอักษร (character) ที่เรียงต่อกัน โดยที่ชนิดข้อมูลนี้ ประกอบด้วยสมาชิกและฟังก์ชันให้สามารถเรียกใช้งานได้ตามตารางดังต่อไปนี้

ตารางที่ 3-1 ฟังก์ชันสำหรับ string ที่ใช้งานบ่อย

Expression	Effect
strVar. at (index)	Return the element at the position specified by index .
strVar[index]	Return the element at the position specified by index .
strVar. append (n, ch)	Appends n copies of ch to strVar , in which ch is a char variable or a char constant.
strVar. append (str)	Appends str to strVar.
strVar. clear ()	Deletes all the characters in strVar.
strVar. compare (str)	Compares strVar and str.
strVar. empty ()	Returns true if strVar is empty; otherwise, it returns false .
strVar. erase ()	Deletes all the characters in strVar.
strVar. erase (pos, n)	Deletes n characters from strVar starting at position pos.
strVar. find (str)	Returns the index of the first occurrence of str in strVar. If str is not found, the special value string::npos is returned.
strVar. find (str, pos)	Returns the index of the first occurrence at or after pos where str is found in strVar
strVar. find_first_of (str, pos)	Returns the index of the first occurrence of any character of strVar in str. The search starts at pos.
strVar. find_first_not_of (str, pos)	Returns the index of the first occurrence of any character of str not in strVar. The search starts at pos.
strVar. insert (pos, n, ch);	Inserts n occurrences of the character ch at index pos into strVar; pos and n are of type string::size_type; ch is a character.
strVar. insert (pos, str);	Inserts all the characters of str at index pos into strVar.
strVar. length ()	Returns a value of type string::size_type giving the number of character strVar.
strVar. replace (pos, n, str);	Starting at index pos, replaces the next n character of strVar with all the characters of str. If n > length of strVar, then all the characters until the end of strVar are replaced.

Expression	Effect
strVar.substr (pos, len)	Returns a string that is a substring of strVar starting at pos. The length of the substring is at most len characters. If len is too large, it means "to the end" of the string in strVar.
strVar.size ()	Returns a value of type string::size_type giving the number of character strVar.
strVar.swap (str1)	Swaps the contents of strVar and str1. str1 is a string variable.

ชนิดตัวแปร string นั้นมีสมาชิกที่น่าสนใจ ดังนี้

string::npos เป็นค่าสูงสุดที่จัดเก็บข้อมูลชนิด string ได้ (โดยค่าขึ้นอยู่กับ compiler และเครื่องคอมพิวเตอร์ นั้น เช่น 18,446,744,073,709,551,615)

1. ให้นักนิสิตสร้างไฟล์ ex02_npos.cpp มี source code ดังต่อไปนี้

```
//*****
// <STUDENT_FULLNAME>
// Example string operations
//*****
#include <iostream>
#include <string>

using namespace std;
int main()
{
    cout << "string npos = " << string::npos << endl;
    string s = "Hello World!";
    if(s.find('a') == string::npos){
        cout << "no 'a' in 'Hello World!'\n";
    }

    int position = s.find('W');
    if(position != string::npos){
        cout << "Character 'W' in " << s
            << "is position : " << position << endl;
    }
    // functions that take string subsets as arguments
    // use npos as the "all the way to the end" indicator
    string s2 = s.substr( 2, string::npos);
    cout << s2 << '\n';
}
```

ผลลัพธ์ของโปรแกรมคือ

.....

.....

.....

.....

.....

.....

2. ให้นักนิสิตสร้างไฟล์ ex03_stringFunction.cpp มี source code ดังต่อไปนี้

```

//*****
// <STUDENT_FULLNAME>
// Example: clear, empty, erase, length, and size functions
//*****

#include <iostream>
#include <string>
using namespace std;
int main() {
    string firstName = "Elizabeth";           //Line 1
    string name = firstName + " Jones";       //Line 2
    string str1 = "It is sunny.";             //Line 3
    string str2 = "";                         //Line 4
    string str3 = "computer science";         //Line 5
    string str4 = "C++ programming.";         //Line 6
    string str5 = firstName + " is taking " + str4; //Line 7
    string::size_type len;                    //Line 8
    cout << "Line 9: str3: " << str3 << endl;  //Line 9
    str3.clear();                             //Line 10
    cout << "Line 11: After clear, str3: " << str3 << endl; //Line 11
    cout << "Line 12: str1.empty(): "
        << str1.empty() << endl;             //Line 12
    cout << "Line 13: str2.empty(): "
        << str2.empty() << endl;             //Line 13
    cout << "Line 14: str4: " << str4 << endl;  //Line 14
    str4.erase(11, 4);                        //Line 15
    cout << "Line 16: After erase(11, 4), str4: "
        << str4 << endl; //Line 16
    cout << "Line 17: Length of \"" << firstName << "\" = "
        << static_cast<unsigned int> (firstName.length())
        << endl;                               //Line 17
    cout << "Line 18: Length of \"" << name << "\" = "
        << static_cast<unsigned int> (name.length())
        << endl;                               //Line 18
    cout << "Line 19: Length of \"" << str1 << "\" = "
        << static_cast<unsigned int> (str1.length())
        << endl;                               //Line 19
    cout << "Line 20: Size of \"" << str5 << "\" = "
        << static_cast<unsigned int> (str5.size())
        << endl;                               //Line 20
    len = name.length();                      //Line 21
    cout << "Line 22: len = "
        << static_cast<unsigned int> (len) << endl; //Line 22
    return 0;                                //Line 23
}

```

ผลลัพธ์ของโปรแกรมคือ

.....

.....

.....

.....

.....

.....

3. จาก Line 15 หากเปลี่ยนจาก str4.erase(11, 4); เป็น str4.erase(1, 2); นิสิตคิดว่า str4 จะมีค่าเท่ากับ

4. ให้นิสิตสร้างไฟล์ ex04_stringFind.cpp มี source code ดังต่อไปนี้ และทดลอง compile และรันโปรแกรม

```

//*****
// <STUDENT_FULLNAME>
// Example: find function
//*****

#include <iostream>
#include <string>
using namespace std;
int main() {
    string sentence = "Outside it is cloudy and warm."; //Line 1
    string str = "cloudy"; //Line 2
    string::size_type position; //Line 3
    cout << "Line 4: sentence = \"" << sentence //Line 4
    << "\"" << endl;
    cout << "Line 5: The position of \"is\" in sentence = "
    << static_cast<unsigned int> (sentence.find("is"))
    << endl; //Line 5
    cout << "Line 6: The position of 's' in sentence = "
    << static_cast<unsigned int> (sentence.find('s'))
    << endl; //Line 6
    cout << "Line 7: The position of \"" << str << "\" in sentence = "
    << static_cast<unsigned int> (sentence.find(str))
    << endl; //Line 7
    cout << "Line 8: The position of \"the\" in sentence = "
    << static_cast<unsigned int> (sentence.find("the"))
    << endl; //Line 8
    cout << "Line 9: The first occurrence of 'i' in "
    << "sentence \n after position 6 = "
    << static_cast<unsigned int> (sentence.find('i', 6))
    << endl; //Line 9
    position = sentence.find("warm"); //Line 10
    cout << "Line 11: " << "Position = " << position
    << endl; //Line 11
    return 0; //Line 12
}

```

ผลลัพธ์ของโปรแกรมคือ

.....

.....

.....

.....

.....

.....

5. ให้นิสิตสร้างไฟล์ ex05_insertAndReplace.cpp สำหรับทดสอบฟังก์ชัน insert และ replace ของ string โดย source code ดังต่อไปนี้ ให้นิสิตทำการทดลอง compile และรันโปรแกรม

```

//*****
//// <STUDENT_FULLNAME>
//// Example: insert and replace functions
////*****

#include <iostream>
#include <string>
using namespace std;
int main() {
    string firstString = "Cloudy and warm.";           //Line 1
    string secondString = "Hello there";               //Line 2
    string thirdString = "Henry is taking programming I."; //Line 3
    string str1 = " very ";                             //Line 4
    string str2 = "Lisa";                               //Line 5
    cout << "Line 6: firstString = " << firstString << endl; //Line 6
    firstString.insert(10, str1);                       //Line 7
    cout << "Line 8: After insert; firstString = " << firstString //Line 8
        << endl;
    cout << "Line 9: secondString = " << secondString << endl; //Line 9
    secondString.insert(11, 5, '!');                   //Line 10
    cout << "Line 11: After insert; secondString = " << secondString //Line 11
        << endl;
    cout << "Line 12: thirdString = " << thirdString << endl; //Line 12
    thirdString.replace(0, 5, str2);                   //Line 13
    cout << "Line 14: After replace, thirdString = " //Line 14
        << thirdString << endl;
    return 0;                                          //Line 15
}

```

ผลลัพธ์ของโปรแกรมคือ

.....

.....

.....

.....

.....

.....

6. ให้นักนิสิตสร้างไฟล์ ex06_substr.cpp มี source code ดังต่อไปนี้ สำหรับทดสอบฟังก์ชัน substr ของ string ให้นักนิสิตทำการทดลอง compile และรันโปรแกรม

```

//*****
//// <STUDENT_FULLNAME>
//// Example: substr function
////*****
#include <iostream>
#include <string>
using namespace std;
int main() {
    string sentence;                                //Line 1
    string str;                                     //Line 2
    sentence = "It is cloudy and warm.";           //Line 3
    cout << "Line 4: substr(0, 5) in \"\" << sentence << "\" = \"\"
        << sentence.substr(0, 5) << "\" << endl;    //Line 4
    cout << "Line 5: substr(6, 6) in \"\" << sentence << "\" = \"\"
        << sentence.substr(6, 6) << "\" << endl;    //Line 5
    cout << "Line 6: substr(6, 16) in \"\" << sentence << "\" = \"\" << "\"
        << sentence.substr(6, 16) << "\" << endl;    //Line 6
    cout << "Line 7: substr(17, 10) in \"\" << sentence << "\" = \"\"
        << sentence.substr(17, 10) << "\" << endl;    //Line 7
    cout << "Line 8: substr(3, 6) in \"\" << sentence << "\" = \"\"
        << sentence.substr(3, 6) << "\" << endl;    //Line 8
    str = sentence.substr(0, 8);                     //Line 9
    cout << "Line 10: \" << \"str = \"\" << str << "\" << endl; //Line 10
    str = sentence.substr(2, 10);                     //Line 11
    cout << "Line 12: \" << \"str = \"\" << str << "\" << endl; //Line 12
    return 0;
}

```

ผลลัพธ์ของโปรแกรมคือ

.....

.....

.....

.....

.....

.....

.....

.....

.....

ตอนที่ 3 การเรียกใช้งานฟังก์ชันในภาษา C++

ใน C++ มี predefined function หรือฟังก์ชันมาตรฐานที่ถูกเขียนไว้แล้วอยู่ในไลบรารีให้นักพัฒนาสามารถเรียกใช้งานได้ เช่น หากต้องการใช้งานฟังก์ชันสำหรับการหาค่าของ 3 ยกกำลัง 5 (3^5) สามารถเรียกใช้ predefined function ที่ชื่อว่า pow ได้ โดยทำการเรียกใช้งาน header ที่ชื่อว่า cmath

```
#include <cmath>
#include <iostream>
using namespace std;
int main()
{
    Cout << "3 power 5 = " << pow(3,5) << endl;
    return 0;
}
```

ตารางที่ 3-2 ฟังก์ชันใน C++ ที่ใช้งานบ่อย

Function	Header File	Purpose	Parameter(s) Type	Result
abs(x)	<cmath>	Returns the absolute value of its arguments: abs(-7) = 7	double	double
ceil(x)	<cmath>	Returns the smallest whole number that is not less than x: ceil(56.34) = 57.0	double	double
cos(x)	<cmath>	Returns the cosine of angle: x: cos(0.0) = 1.0	double (radians)	double
exp(x)	<cmath>	Return e^x , where = 2.718: exp(1.0) = 2.71828	double	double
fabs(x)	<cmath>	Returns the absolute value of its argument: fabs(-5.67) = 5.67	double	double
floor(x)	<cmath>	Return the largest whole number that is not greater than x: floor(45.67) = 45.00	double	double
islower(x)	<cctype>	Returns 1 (true) if x is a lowercase letter; otherwise, it returns 0 (false); islower('h') is 1 (true)	int	int

Function	Header File	Purpose	Parameter(s) Type	Result
isupper(x)	<cctype>	Returns 1 (true) if x is an uppercase letter; otherwise, it returns 0 (false); isupper('K') is 1 (true)	int	int
pow(x, y)	<cmath>	Return x^y ; if x is negative, y must be a whole number: pow(0.16, 0.5) = 0.4	double	double
sqrt(x)	<cmath>	Returns the nonnegative square root of x; x must be nonnegative: sqrt(4.0) = 2.0	double	double
tolower(x)	<cctype>	Returns the lowercase value of x if x is uppercase; otherwise, it returns x	int	int
toupper(x)	<cctype>	Returns the uppercase value of x if x is lowercase; otherwise, it returns x	int	int

1. ให้นักนิสิตสร้างไฟล์ ex07_predefinedFunction.cpp มี source code ดังต่อไปนี้ สำหรับทดสอบการเรียกใช้ฟังก์ชัน แล้วทำการทดลอง compile และรันโปรแกรม

```
//*****8*****
// <STUDENT_FULLNAME>
// How to use predefined functions.
//*****

#include <iostream>
#include <cmath>
#include <cctype>
using namespace std;
int main() {
    double x;
    double u, v;
    cout << "Line 1: Uppercase a is "
         << static_cast<char>(toupper('a'))
         << endl; // Line 1
    u = 4.2; // Line 2
    v = 3.0; // Line 3
    cout << "Line 4: " << u << " to the power of "
         << v << " = " << pow(u, v) << endl; // Line 4
    cout << "Line 5: 5.0 to the power of 4 = "
         << pow(5.0, 4) << endl; // Line 5
    u = u + pow(3.0, 3); // Line 6
    cout << "Line 7: u = " << u << endl; // Line 7
    x = -15.3; // Line 8
    cout << "Line 9: Absolute value of " << x
         << " = " << abs(x) << endl; // Line 9
    return 0;
}
```

2. ในโปรแกรม ex07_predefinedFunction.cpp ฟังก์ชันใดบ้างอยู่ในไลบรารี cmath

.....

3. ในโปรแกรม ex07_predefinedFunction.cpp ฟังก์ชันใดบ้างอยู่ในไลบรารี ctype

.....

ตอนที่ 4 การสร้างฟังก์ชันใน C++

รูปแบบการประกาศฟังก์ชันคือ

```
functionType functionName (formal parameter list) {
    ...statement...
}
```

โดยมีรายละเอียดคือ

functionType คือ ชนิดของค่าที่จะส่งกลับจากฟังก์ชันเมื่อทำงานเรียบร้อยแล้ว
functionName คือ ชื่อของฟังก์ชัน

รูปแบบ formal parameter list คือ

```
dataType identifier, [dataType identifier, ...]
```

โดยมีรายละเอียดดังนี้

dataType คือ ชนิดของตัวแปรที่ส่งค่าเข้าสู่ฟังก์ชัน
identifier คือ ชื่อตัวแปรรับค่ามาใช้งานในฟังก์ชัน

รูปแบบการคืนค่าในฟังก์ชัน คือ

```
return expr;
```

โดยมีรายละเอียดดังนี้

return เป็นคำเฉพาะในการคืนค่ากลับ
expr คือ ตัวแปรหรือค่าที่ต้องการคืนออกจากฟังก์ชัน

1. ให้นิสิตทำการสร้างไฟล์ ex08_userDefinedFunction.cpp สำหรับทดสอบการสร้างฟังก์ชัน larger และ compareThree โดยมี source code ดังต่อไปนี้

```

//*****
// <STUDENT_FULLNAME>
// Program: Large of numbers
//*****

using namespace std;
#include <iostream>
double larger(double x, double y);
double compareThree(double x, double y, double z);
int main() {
    double one, two; // Line 1
    cout << "Line 2: The larger of 5 and 10 is "
         << larger(5, 10) << endl; // Line 2
    cout << "Line 3: Enter two numbers: "; // Line 3
    cin >> one >> two; // Line 4
    cout << "Line 5: The larger of " << one
         << " and " << two << " is "
         << larger(one, two) << endl; // Line 5
    cout << "Line 7: The larger of 43.48, 34.00 " // Line 6
         << "and 12.65 is "
         << compareThree(43.48, 34.00, 12.65)
         << endl; // Line 7
    return 0;
}

```

```
double larger (double x, double y) {
    double max;
    if(x >= y)
        max = x;
    else
        max = y;
    return max;
}
double compareThree(double x, double y, double z) {
    return larger(x, larger(y, z));
}
```

2. ให้นิสิตตอบคำถามดังต่อไปนี้

ฟังก์ชัน larger มีพารามิเตอร์กี่ตัว ชนิดใดบ้าง และมีการคืนค่าชนิดใด

Identifier parameter (ชื่อตัวแปรพารามิเตอร์)	ชนิดข้อมูล

ฟังก์ชัน larger มีการคืนค่าชนิดใด

.....

ตอนที่ 5 การสร้างฟังก์ชันที่ไม่มีการคืนค่า (Void)

การสร้างฟังก์ชันโดยไม่มีการคืนค่า มีรูปแบบดังนี้

```
void functionName (formal parameter list) {
    ... statements ...
}
```

มีรายละเอียดดังนี้

void คือ คำเฉพาะในการกำหนดฟังก์ชันที่ไม่มีการคืนค่า

รูปแบบ formal parameter list คือ

dataType identifier, [dataType identifier, ...]

โดยมีรายละเอียดดังนี้

dataType คือ ชนิดของตัวแปรที่ส่งค่าเข้าสู่ฟังก์ชัน

identifier คือ ชื่อตัวแปรรับค่ามาใช้งานในฟังก์ชัน

ในฟังก์ชันที่ไม่มีการคืนค่า จะไม่มีการเรียกใช้ return statement

1. ให้นิสิตสร้างไฟล์ ex09_void.cpp สำหรับทดลองสร้างฟังก์ชันที่ไม่มีการคืนค่า โดยมี source code ดังนี้

```
/**
 * *****
 * // < STUDENT_FULLNAME >
 * // Program: Print a triangle of stars
 * *****
 */
#include <iostream>
using namespace std;
void printStars(int blanks, int starsInLine);
int main() {
    int noOfLines;           //variable to store the number of lines
    int counter;             //for loop control variable
```

```

int noOfBlanks;          //variable to store the number of blanks
cout << "Enter the number of star lines (1 to 20) "
    << "to be printed: ";           //Line 1
cin >> noOfLines;                //Line 2
while (noOfLines < 0 || noOfLines > 20) {           //Line 3
    cout << "Number of star lines should be "
        << "between 1 and 20"<<endl;           //Line 4
    cout << "Enter the number of star lines "
        << "(1 to 20) to be printed: ";           //Line 5
    cin >> noOfLines;           //Line 6
}
cout << endl << endl;           //Line 7
noOfBlanks = 30; //Line 8
for (counter = 1; counter <= noOfLines; counter++) { //Line 9
    printStars(noOfBlanks, counter);           //Line 10
    noOfBlanks--;           //Line 11
}
return 0;           //Line 12
}

void printStars(int blanks, int starsInLine) {
    int count;
    for (count = 1; count <= blanks; count++)           //Line 13
        cout << ' ';           //Line 14
    for (count = 1; count <= starsInLine; count++)           //Line 15
        cout << " *";           //Line 16
    cout << endl;
}

```

ตอนที่ 6 การสร้างฟังก์ชันแบบ pass by value

การส่งพารามิเตอร์แบบ pass by value เป็นการส่งพารามิเตอร์โดยการคัดลอกค่าในตัวแปรเพื่อส่งเข้าสู่ฟังก์ชัน ดังนั้นเมื่อค่าของตัวแปรในฟังก์ชันเปลี่ยนแปลงไป หลังการทำงานฟังก์ชันเรียบร้อยแล้ว ค่าตัวแปรเดิมที่ถูกส่งเข้าในฟังก์ชันจะไม่ถูกเปลี่ยนแปลงตามไปด้วย

1. ให้นิสิตสร้างไฟล์ ex10_passByValue.cpp สำหรับทดลองสร้างฟังก์ชันแบบ pass by value โดยมี source code ดังนี้

```
//*****
// < STUDENT_FULLNAME >
// Program: how a value parameter works.
//*****
#include <iostream>
using namespace std;
void funcValueParam(int num);
int main() {
    int number = 6;                                     //Line 1
    cout << "Line 2: Before calling the function "
         << "funcValueParam, number = " << number
         << endl;                                       //Line 2
    funcValueParam(number);                             //Line 3
    cout << "Line 4: After calling the function "
         << "funcValueParam, number = " << number
         << endl;                                       //Line 4
    return 0;
}
void funcValueParam(int num) {
    cout << "Line 5: In the function funcValueParam, "
         << "before changing, num = " << num
         << endl;                                       //Line 5
    num = 15;                                           //Line 6
    cout << "Line 7: In the function funcValueParam, "
         << "after changing, num = " << num
         << endl;                                       //Line 7
}
```

2. จากการเรียกฟังก์ชัน funcValueParam ที่ Line 3 แล้ว ค่า number ที่ Line 4 มีการเปลี่ยนแปลงค่าหรือไม่ เพราะเหตุใด

.....

.....

.....

.....

ตอนที่ 7 การสร้างฟังก์ชันแบบ pass by reference

การส่งพารามิเตอร์แบบ pass by reference เป็นการส่งแอดเดรสในหน่วยความจำของตัวแปรเข้าสู่ฟังก์ชัน ทำให้เมื่อมีการเปลี่ยนแปลงค่าของตัวแปรที่ส่งค่าแบบนี้จะทำให้มีการเปลี่ยนแปลงค่าตามแอดเดรสที่อยู่ในหน่วยความจำ ส่งผลให้เมื่อทำงานฟังก์ชันเรียบร้อยแล้ว ค่าเดิมที่ส่งเข้าไปในฟังก์ชันจะเปลี่ยนแปลงไปด้วย

1. ให้นิสิตสร้างไฟล์ ex11_passByRefernce.cpp สำหรับทดลองสร้างฟังก์ชันแบบ pass by reference โดยมี source code ดังนี้

```
//*****
// < STUDENT_FULLNAME >
// Program: A pass by reference
//*****

#include <iostream>
using namespace std;
void getScore(int& score);
void printGrade(int score);
int main () {
    int courseScore;
    cout << "Line 1: Based on the course score, \n"
        << " this program computes the "
        << "course grade." << endl;           //Line 1
    getScore(courseScore);                     //Line 2
    printGrade(courseScore);                   //Line 3
    return 0;
}
void getScore(int& score) {
    cout << "Line 4: Enter course score: ";    //Line 4
    cin >> score;                             //Line 5
    cout << endl << "Line 6: Course score is "
        << score << endl;                     //Line 6
}
void printGrade(int cScore) {
    cout << "Line 7: Your grade for the course is "; //Line 7
    if (cScore >= 90)                             //Line 8
        cout << "A." << endl;
    else if (cScore >= 80)
        cout << "B." << endl;
    else if (cScore >= 70)
        cout << "C." << endl;
    else if (cScore >= 60)
        cout << "D." << endl;
    else
        cout << "F." << endl;
}
```

2. ฟังก์ชัน getScore เป็นฟังก์ชันที่มีการส่งพารามิเตอร์แบบ
3. ฟังก์ชัน printGrade เป็นฟังก์ชันที่มีการส่งพารามิเตอร์แบบ

ให้นักนิสิตสร้างไฟล์ ex12_passByValueAndReference.cpp สำหรับทดลองสร้างฟังก์ชันแบบ pass by value และ pass by reference โดยมี source code ดังนี้

```

//*****
// < STUDENT_FULLNAME >
// Program: Reference and value parameters.
//*****
#include <iostream>
using namespace std;
void addFirst(int& first, int& second);
void doubleFirst(int one, int two);
void squareFirst(int& ref, int val);
int main () {
    int num = 5;
    cout << "Line 1: Inside main: num = " << num
        << endl; //Line 1
    addFirst(num, num); //Line 2
    cout << "Line 3: Inside main after addFirst:"
        << " num = " << num << endl; //Line 3
    doubleFirst(num, num); //Line 4
    cout << "Line 5: Inside main after "
        << "doubleFirst: num = " << num << endl; //Line 5
    squareFirst(num, num); //Line 6
    cout << "Line 7: Inside main after "
        << "squareFirst: num = " << num << endl; //Line 7
    return 0;
}
void addFirst(int& first, int& second) {
    cout << "Line 8: Inside addFirst: first = "
        << first << ", second = " << second << endl; //Line 8
    first = first + 2; //Line 9
    cout << "Line 10: Inside addFirst: first = "
        << first << ", second = " << second << endl; //Line 10
    second = second * 2; //Line 11
    cout << "Line 12: Inside addFirst: first = "
        << first << ", second = " << second << endl; //Line 12
}
void doubleFirst(int one, int two) {
    cout << "Line 13: Inside doubleFirst: one = "
        << one << ", two = " << two << endl; //Line 13
    one = one * 2; //Line 14
    cout << "Line 15: Inside doubleFirst: one = "
        << one << ", two = " << two << endl; //Line 15
    two = two + 2; //Line 16
    cout << "Line 17: Inside doubleFirst: one = "
        << one << ", two = " << two << endl; //Line 17
}
void squareFirst(int& ref, int val) {
    cout << "Line 18: Inside squareFirst: ref = "
        << ref << ", val = " << val << endl; //Line 18
    ref = ref * ref; //Line 19
    cout << "Line 20: Inside squareFirst: ref = "
        << ref << ", val = " << val << endl; //Line 20
    val = val + 2; //Line 21
    cout << "Line 22: Inside squareFirst: ref = "
        << ref << ", val = " << val << endl; //Line 22
}

```

ฟังก์ชัน squareFirst มีการส่งพารามิเตอร์ชื่ออะไร และแบบใดบ้าง

Identifier parameter (ชื่อตัวแปรพารามิเตอร์)	ชนิดข้อมูล	การส่งค่าแบบ (pass by value หรือ pass by reference)

ฟังก์ชัน squareFirst มีการคืนค่าชนิดใด

.....

ตอนที่ 8 การกำหนดค่าเริ่มต้นให้กับพารามิเตอร์ของฟังก์ชัน

โดยปกติการเรียกใช้ฟังก์ชันจำเป็นต้องส่งพารามิเตอร์ให้เท่ากับจำนวนที่ประกาศไว้ตอนสร้างฟังก์ชัน แต่ C++ สามารถส่งพารามิเตอร์น้อยกว่าจำนวนพารามิเตอร์ที่ประกาศตอนสร้างฟังก์ชันได้ เพียงแต่ต้องกำหนดค่าเริ่มต้นให้กับพารามิเตอร์ไว้ก่อน และตำแหน่งของพารามิเตอร์ต้องตรงกัน

1. ให้นิสิตสร้างไฟล์ ex13_default.cpp สำหรับทดลองกำหนดค่าเริ่มต้นให้กับพารามิเตอร์ของฟังก์ชัน โดยมี source code ดังนี้

```
//*****
// < STUDENT_FULLNAME >
// Program: default value
//*****
#include <iostream>
#include <iomanip>
using namespace std;
int volume(int l = 1, int w = 1, int h = 1);
void funcOne(int& x, double y = 12.34, char z = 'B');
int main() {
    int a = 23;
    double b = 48.78;
    char ch = 'M';
    cout << fixed << showpoint;
    cout << setprecision(2);
    cout << "Line 1: a = " << a << ", b = "
        << b << ", ch = " << ch << endl;           //Line 1
    cout << "Line 2: Volume = " << volume()
        << endl;                                     //Line 2
    cout << "Line 3: Volume = " << volume(5, 4)
        << endl;                                     //Line 3
    cout << "Line 4: Volume = " << volume(34)
        << endl; //Line 4
    cout << "Line 5: Volume = "
        << volume(6, 4, 5) << endl;                 //Line 5
    funcOne(a); //Line 6
    funcOne(a, 42.68);                             //Line 7
    funcOne(a, 34.65, 'Q');                         //Line 8
    cout << "Line 9: a = " << a << ", b = "
        << b << ", ch = " << ch << endl;           //Line 9
    return 0;
}
```

```
int volume(int l, int w, int h) {
    return l * w * h;                                     //Line 10
}
void funcOne(int& x, double y, char z) {
    x = 2 * x; //Line 11
    cout << "Line 12: x = " << x << ", y = "
        << y << ", z = " << z << endl;                //Line 12
}
```

ฟังก์ชัน volume มีการส่งพารามิเตอร์ชื่ออะไร แบบใดบ้าง และค่าตั้งต้นคือค่าใด

Identifier parameter (ชื่อตัวแปรพารามิเตอร์)	ชนิดข้อมูล	การส่งค่าแบบ (pass by value หรือ pass by reference)	ค่าตั้งต้น

ผลลัพธ์ของโปรแกรม

Line 2: Volume มีค่าเท่ากับ1.....เกิดจากการเรียกใช้ฟังก์ชันvolume().....
ภายในฟังก์ชัน volume ได้ทำการคำนวณค่า 1 * 1 * 1.....แล้วส่งค่ากลับมาที่ main program
Line 3: Volume มีค่าเท่ากับเกิดจากการเรียกใช้ฟังก์ชัน
ภายในฟังก์ชัน volume ได้ทำการคำนวณค่าแล้วส่งค่ากลับมาที่ main program
Line 4: Volume มีค่าเท่ากับเกิดจากการเรียกใช้ฟังก์ชัน
ภายในฟังก์ชัน volume ได้ทำการคำนวณค่าแล้วส่งค่ากลับมาที่ main program
Line 5: Volume มีค่าเท่ากับเกิดจากการเรียกใช้ฟังก์ชัน
ภายในฟังก์ชัน volume ได้ทำการคำนวณค่าแล้วส่งค่ากลับมาที่ main program

ตอนที่ 9 โอเวอร์โหลดฟังก์ชัน (overloaded function)

โอเวอร์โหลดฟังก์ชัน(overloaded function) เป็นฟังก์ชันที่สามารถทำงานได้หลายๆ ลักษณะขึ้นอยู่กับพารามิเตอร์ที่ผู้ใช้กำหนดให้

วิธีการสร้าง overloaded function มีวิธีการสร้างเหมือนกับฟังก์ชันปกติ โดยที่มีการสร้างชื่อฟังก์ชันเหมือนกัน แต่กำหนดพารามิเตอร์แต่ละฟังก์ชันให้แตกต่างกัน

ให้นิสิตทำการสร้างไฟล์ ex14_overloadFunction.cpp สำหรับการทำ overload function ฟังก์ชัน print โดยมี source code ดังต่อไปนี้

```
//*****
// < STUDENT_FULLNAME >
// Program: overload function
//*****
#include <iostream>
void print(int i);
void print(double f);
void print(char* c) ;
```

```
using namespace std;
int main()
{
    int int_number_1 = 30;
    double double_number_1 = -12.3;
    char ch = 'A';

    print(int_number_1);
    print(double_number_1);
    print(ch);

    return 0;
}

void print(int i) {
    cout << "Printing int: " << i << endl;
}
void print(double f) {
    cout << "Printing float: " << f << endl;
}
void print(char* c) {
    cout << "Printing character: " << c << endl;
}
```

ให้นักนิสิตสังเกตว่าฟังก์ชัน print มีพารามิเตอร์อะไรบ้าง

Identifier parameter (ชื่อตัวแปรพารามิเตอร์)	ชนิดข้อมูล	การคืนค่า

รหัสனிสิต..... ชื่อ สกุล กลุ่ม

ฟังก์ชัน distance

.....

.....

.....

.....

.....

.....

.....

.....

ฟังก์ชัน radius

.....

.....

.....

.....

.....

.....

.....

.....

ฟังก์ชัน circumference

.....

.....

.....

.....

.....

.....

.....

.....

ฟังก์ชัน area

.....

.....

.....

.....

.....

.....

.....

.....