

บทที่ 5 ฟังก์ชัน (Function)

วัตถุประสงค์

1. เพื่อให้นักศึกษาเข้าใจหลักการทำงานของฟังก์ชันมาตรฐานของ C++
2. เพื่อให้นักศึกษาเข้าใจหลักการทำงานของฟังก์ชัน และตัวแปรแบบต่างๆ
3. เพื่อให้นักศึกษาสามารถสร้างฟังก์ชันและใช้ตัวแปร รวมทั้งการส่งผ่านค่าของตัวแปรไปหฟังก์ชันได้

ฟังก์ชัน

สำหรับโปรแกรมภาษา C++ นั้น การทำงานจะอยู่ในลักษณะการเรียกใช้ฟังก์ชันเป็นหลัก โดยภาษา C++ สามารถเรียกใช้ฟังก์ชันของระบบหรือสร้างฟังก์ชันเองได้ และสามารถใช้งานได้มากกว่าหนึ่งฟังก์ชัน แต่ในโปรแกรมหนึ่งๆ จะมีฟังก์ชัน main() ได้เพียงฟังก์ชันเดียวเท่านั้น ฟังก์ชันสามารถแบ่งออกได้เป็น 2 ประเภทคือ

1. ฟังก์ชันมาตรฐาน (Standard Libraries Function)
2. ฟังก์ชันผู้กำหนด (User-Define Function)

1. ฟังก์ชันมาตรฐาน (Standard Libraries Function)

เป็นการใช้งานฟังก์ชันของภาษา C++ ที่ได้ถูกกำหนดไว้แล้วในไลบรารีมาตรฐาน ซึ่งมีฟังก์ชันต่างๆ อย่างมากมาย โดยจะแบ่งการทำงานของฟังก์ชันตามชื่อไลบรารีในการใช้งาน ที่สำคัญดังนี้

cctype.h or ctype ปัจจุบันได้รวมกับ namespace std

int isalnum (int Character); ex: isalnum('a') → value > 0	เป็นฟังก์ชันการตรวจสอบว่าเป็นตัวเลขหรือตัวอักษรโดยจะส่งค่าคืนเป็นตัวเลข ถ้าหากเป็นตัวเลขหรือตัวอักษรจะคืนค่าที่ไม่ใช่ 0 และหากเป็นตัวอักษรอื่นจะได้ค่าเป็น 0
int isalpha (int character); ex: isalpha('a') → value > 0	เป็นฟังก์ชันการตรวจสอบว่าเป็นตัวอักษรโดยจะส่งค่าคืนเป็นตัวเลข ถ้าหากเป็นตัวอักษรจะคืนค่าที่ไม่ใช่ 0 และหากไม่เป็นตัวอักษรจะได้ค่าเป็น 0
int isdigit (int character); ex: isdigit('9') → value > 0	เป็นฟังก์ชันการตรวจสอบว่าเป็นตัวเลข โดยจะส่งค่าคืนเป็นตัวเลข ถ้าหากเป็นตัวเลขจะคืนค่าที่ไม่ใช่ 0 และหากไม่เป็นตัวเลขจะได้ค่าเป็น 0
int islower (int character); ex: islower('a') → value > 0	เป็นฟังก์ชันการตรวจสอบว่าเป็นตัวอักษรเล็ก โดยจะส่งค่าคืนเป็นตัวเลข ถ้าหากเป็นตัวอักษรเล็กจะคืนค่าที่ไม่ใช่ 0 และหากไม่เป็นตัวอักษรเล็กจะได้ค่าเป็น 0
int isspace (int character); ex: isspace(' ') → value > 0	เป็นฟังก์ชันการตรวจสอบว่าเป็นช่องว่าง โดยจะส่งค่าคืนเป็นตัวเลข ถ้าหากเป็นช่องว่างจะคืนค่าที่ไม่ใช่ 0 และหากไม่เป็นช่องว่างจะได้ค่าเป็น 0
int isupper (int character); ex: isupper('D') → value > 0	เป็นฟังก์ชันการตรวจสอบว่าเป็นตัวอักษรใหญ่ โดยจะส่งค่าคืนเป็นตัวเลข ถ้าหากเป็นตัวอักษรใหญ่จะคืนค่าที่ไม่ใช่ 0 และหากไม่เป็นตัวอักษรใหญ่จะได้ค่าเป็น 0

int tolower (int Character); ex: tolower('a') → 'A'	เป็นฟังก์ชันการแปลงตัวอักษรใหญ่เป็นตัวอักษรเล็ก โดยจะส่งค่าคืนเป็นตัวอักษรเล็ก
int toupper (int Character); ex: toupper('A') → 'a'	เป็นฟังก์ชันการแปลงตัวอักษรเล็กเป็นตัวอักษรใหญ่ โดยจะส่งค่าคืนเป็นตัวอักษรใหญ่

การทดลองที่ 5_1 การใช้งานฟังก์ชันกับตัวอักษรใน Library ctype

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main()
5  {
6      char Ch;
7      string Message;
8      cout << "Enter character : ";
9      cin >> Ch;
10     cout << endl;
11     if (isalnum(Ch)) {
12         if (isalpha(Ch)) {
13             if (islower(Ch)) Message = "lower character.";
14             else Message = "upper character.";
15         }
16         else if (isdigit(Ch)) Message = "digit.";
17     }
18     else Message = "special character.";
19     cout << "\"" << Ch << "\"" << " is " << Message << endl;
20     if (isalpha(Ch)) {
21         if (islower(Ch)) {
22             cout << "\"" << Ch << "\"" << " convert to upper \";
23             cout << (char)toupper(Ch) << "\"";
24         }
25         else {
26             cout << "\"" << Ch << "\"" << " to lower \";
27             cout << (char)tolower(Ch) << "\"";
28         }
29         cout << endl;
30     }
31     return(0);
32 }
```

บันทึกผลการทดลอง

Stdlib Library (stdlib.h or cstdlib) ปัจจุบันได้รวมกับ namespace std

double atof (const char *string) ; ex: atof("45.6") → 45.6	เป็นฟังก์ชันการแปลงค่าข้อความเป็นตัวเลทศนิยม โดยส่งค่าคืนเป็นตัวเลทศนิยมแบบ double
int atoi (const char *string) ; ex: atof("30") → 30	เป็นฟังก์ชันการแปลงค่าข้อความเป็นตัวเลทจำนวนเต็ม โดยส่งค่าคืนเป็นตัวเลทจำนวนเต็ม int
long int atol(const char *string) ; ex: atol("100000") → 100000	เป็นฟังก์ชันการแปลงค่าข้อความเป็นตัวเลทจำนวนเต็ม โดยส่งค่าคืนเป็นตัวเลทจำนวนเต็มแบบ long int
int rand (void) ; ex: rand () → ?	เป็นฟังก์ชันการสุ่มค่าตัวเลทในช่วง 0 to RAND_MAX, โดยมีค่า seed ในการสร้างตัวเลทสุ่มคือ 1 โดยส่งค่าคืนเป็นตัวเลทจำนวนเต็มแบบ int
void srand (unsigned int Seed) ; ex: srand (100) value > 0	เป็นฟังก์ชันการกำหนดค่า Seed ซึ่งจะเป็นค่าที่จะใช้ในการสุ่มตัวเลทของฟังก์ชัน rand
int abs (int x) ; ex: abs (55) or abs (-55) → 55	เป็นฟังก์ชันการหาค่าสัมบูรณ์ของค่า x โดยส่งค่าคืนเป็นตัวเลทจำนวนเต็ม

การทดลองที่ 5_2 การใช้งานฟังก์ชันใน Library cstdlib

1	#include <iostream>
2	#include <string>
3	using namespace std;
4	int main()
5	{
6	int IntValue;
7	long LongValue;
8	float FloatValue;
9	string StrValue;
10	cout << "Enter string number : ";
11	cin >> StrValue;
12	// convert string to numeric
13	IntValue = atoi(StrValue.c_str());
14	LongValue = atol(StrValue.c_str());
15	FloatValue = atof(StrValue.c_str());
16	cout << endl;
17	cout << "Convert String to Numeric.\n";
18	cout << "*****\n";
19	cout << "Convert to integer = " << IntValue << endl;
20	cout << "Convert to long = " << LongValue << endl;
21	cout << "Convert to float = " << FloatValue << endl << endl;
22	// init seed value to ranom
23	srand(IntValue);
24	cout << "Now random integer number 10 number." << endl;
25	cout << "*****" << endl;
26	for(int N = 1, Num ; N <= 10 ; N++) {
27	Num = rand() % 10;
28	cout << Num << " ";

```
29     }
30     cout << endl;
31     return(0);
32 }
```

บันทึกผลการทดลอง

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Math Library (math.h or cmath) ปัจจุบันได้รวมกับ namespace std

double ceil (double x) ex: ceil (54.3) → 55.0	เป็นฟังก์ชันการหาค่าตัวเลขเพดานบนของค่า x โดยส่งค่าคืนเป็นตัวเลขทศนิยม
double floor (double x) ex: floor (54.3) → 54.0	เป็นฟังก์ชันการหาค่าตัวเลขเพดานล่างของค่า x โดยส่งค่าคืนเป็นตัวเลขทศนิยม
double sqrt (double x) ex: sqrt (4) → 2.0	เป็นฟังก์ชันการหาค่ารากที่สองของค่า x โดยส่งค่าคืนเป็นตัวเลขทศนิยม โดยที่ $x \geq 0$
double exp (double x) ex: exp (2) → 7.389056	เป็นฟังก์ชันการหาค่า exponential ของค่า x ด้วยฐาน e, โดยที่ e มีค่า 2.718282 โดยส่งค่าคืนเป็นตัวเลขทศนิยม
double fabs (double x) ex: fabs (78.5) or fabs (-78.5) → 78.5	เป็นฟังก์ชันการหาค่าสัมบูรณ์ของค่า x โดยส่งค่าคืนเป็นตัวเลขทศนิยม
double log (double x) ex: log (10) → 2.302585	เป็นฟังก์ชันการหา natural logarithm ของค่า x โดยส่งค่าคืนเป็นตัวเลขทศนิยม
double log10 (double x) ex: log10 (10) → 1.0	เป็นฟังก์ชันการหาค่า logarithm ฐานสิบของค่า x โดยส่งค่าคืนเป็นตัวเลขทศนิยม

double pow (double x, double y) ex: pow (5 , 3) → 125.0	เป็นฟังก์ชันการหาค่ายกกำลัง ซึ่ง x เป็นค่าฐานและ y เป็นตัวชี้กำลัง โดยส่งค่าคืนเป็นตัวเลขทศนิยม
double sin (double x) ex: sin (0) → 1	เป็นฟังก์ชันการหาค่า sin ของค่า x , โดยที่ x เป็นค่า radians โดยส่งค่าคืนเป็นตัวเลขทศนิยม
double cos (double x) ex: cos (0) → 1	เป็นฟังก์ชันการหาค่า cosine ของค่า x , โดยที่ x เป็นค่า x เป็น radians โดยส่งค่าคืนเป็นตัวเลขทศนิยม
double tan (double x) ex: tan (0) → 0	เป็นฟังก์ชันการหาค่า tangent ของค่า x , โดยที่ x เป็นค่า x เป็น radians โดยส่งค่าคืนเป็นตัวเลขทศนิยม

การทดลองที่ 5_3 การใช้งานฟังก์ชันใน Library cmath

1	#include <iostream>
2	using namespace std;
3	int main()
4	{
5	double Value;
6	cout << "Enter floating number : ";
7	cin >> Value;
8	cout << endl;
9	cout << "\nCeiling of " << Value << " is " << ceil(Value);
10	cout << "\nFloor of " << Value << " is " << floor(Value);
11	cout << "\nSquare root of " << Value << " is " << sqrt(Value);
12	cout << "\nExponential of " << Value << " is " << exp(Value);
13	cout << "\nFloating absolute of " << Value << " is ";
14	cout << fabs(Value);
15	cout << "\nNatural logarithm of " << Value << " is ";
16	cout << log(Value);
17	cout << "\nLogarithm(10 base) of " << Value << " is ";
18	cout << log10(Value);
19	cout << "\nPower three of " << Value << " is ";
20	cout << pow(Value,3);
21	cout << "\nSin of " << Value << " is " << sin(Value);
22	cout << "\nCosine of " << Value << " is " << cos(Value);
23	cout << "\nTangent of " << Value << " is " << tan(Value);
24	cout << endl;
25	return(0);
26	}

บันทึกผลการทดลอง

String Library (string.h or cstring) ปัจจุบันได้รวมกับ namespace std

char *strcat (char *String1, const char *String2); ex: strcat("Funct", "ions") → "Functions"	เป็นฟังก์ชันการต่อข้อความ โดยส่งค่าคืนเป็นข้อความ
char *strchr (const char *String2, int character); ex: strchr("Funct", 'n') → "nct"	เป็นฟังก์ชันการดึงค่าข้อความตามตำแหน่งตัวอักษรที่กำหนด โดยส่งค่าคืนเป็นข้อความ แต่หากไม่พบตัวอักษรที่กำหนดจะส่งค่าเป็น Null
int strcmp (const *String1, const char *String2); ex: strcmp("TurboC","TurboC") → 0	เป็นฟังก์ชันการเปรียบเทียบข้อความ โดยส่งค่าคืนเป็นตัวเลข คือ 0 หมายถึงข้อความเท่ากัน, 1 หมายถึงข้อความแรกมากกว่า และ -1 หมายถึงข้อความน้อยกว่า
char *strcpy (char *String1, const char *String2); ex: strcpy(str,"STRINGS") → "STRINGS"	เป็นฟังก์ชันการสำเนาข้อความ โดยส่งค่าคืนเป็นข้อความ
size_t strlen (const char *String); ex: strlen ("WRITE") → 5	เป็นฟังก์ชันการหาความยาวของข้อความ โดยส่งค่าคืนเป็นตัวเลขจำนวนเต็ม
char *strncat(char *String1,const char *String2,size_t n); ex: strncat ("Micro", "software",4) → "Microsoft"	เป็นฟังก์ชันการต่อข้อความ ตามจำนวนอักขระที่จะต่อกัน โดยส่งค่าคืนเป็นข้อความ
int strncmp (const *String1,const char *String2,size_t n); ex: strncmp ("TurboC", "Turboc",4) → 0	เป็นฟังก์ชันการเปรียบเทียบข้อความ ตามจำนวนตัวอักษรที่กำหนด โดยส่งค่าคืนเป็นตัวเลข คือ 0 หมายถึงข้อความเท่ากัน, 1 หมายถึงข้อความแรกมากกว่า และ -1 หมายถึงข้อความน้อยกว่า
char *strncpy(char *String1,const char *String2,size_t n); ex: strncpy (str, "WRTIER",5) → "WRITE"	เป็นฟังก์ชันการสำเนาข้อความตามจำนวนตัวอักษรที่กำหนด โดยส่งค่าคืนเป็นข้อความ

การทดลองที่ 5_4 การใช้งานฟังก์ชันใน Library cstring

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      char Str1[30],Str2[30];
6      cout << "Enter string 1 : ";
7      cin >> Str1;
8      cout << "Enter string 2 : ";
9      cin >> Str2;
10     cout << endl;
11     cout << "strcat(Str1,Str2) = " << strcat( Str1, Str2);
12     cout << endl;
13     cout << "strchr(Str1,'s') = " << strchr( Str1, 's') << endl;
14     cout << "strcmp(Str1,Str2) = " << strcmp( Str1, Str2);
15     cout << endl;
16     cout << "strcpy(Str1,Str2) = " << strcpy( Str1, Str2);
17     cout << endl;
18     cout << "strlen(Str1) = " << strlen( Str1) << endl;
19     cout << "strlen(Str2) = " << strlen( Str2) << endl;
20     cout << "String 1 : " << Str1 << endl;
21     cout << "String 1 reverse : " ;
22     for(int N = strlen( Str1) - 1 ; N >= 0 ; N--)
23         cout << Str1[N];
24     cout << endl << endl;
25     return(0);
26 }
```

บันทึกผลการทดลอง

IOMANIP Library (iomanip)

setw(int w)	เป็นการกำหนดความกว้างของข้อความที่จะแสดงตามค่าของ w
setfill(int c)	เป็นการเติมค่าตัวอักษร c ในส่วนที่เป็นช่องว่าง
left	จัดผลลัพธ์ให้ชิดซ้าย ใช้ร่วมกับฟังก์ชัน setw
right	จัดผลลัพธ์ให้ชิดขวา ใช้ร่วมกับฟังก์ชัน setw
setbase(int b)	ใช้กำหนดการแสดงผลจำนวนเต็มเป็นเลขฐาน 10 หรือเลขฐาน 8 หรือเลขฐาน 16
fixed	แสดงจำนวนจริงเป็นเลขจุดทศนิยม 6 หลัก โดยดีฟอลต์ เช่น 123.4 จะแสดงเป็น 123.400000
scientific	แสดงจำนวนจริงเป็นเลขยกกำลัง โดยใช้ตัว e เช่น 123.4 จะแสดงเป็น 1.234 e+02
showpoint	แสดงจำนวนจริงเป็นเลขจุดทศนิยม เช่น 1000 จะแสดงเป็น 1000.00
noshowpoint	ยกเลิกการเซต showpoint
setprecision(int d)	ใช้กำหนดจำนวนหลักของเลขนัยสำคัญสำหรับตัวเลขจำนวนจริงเท่านั้น
skipws	ยกเลิกตัวอักษรที่เป็น WhiteSpace ในขณะรับข้อมูล
noskipws	กำหนดตัวอักษรที่เป็น WhiteSpace ในขณะรับข้อมูล
showpos	แสดงจำนวนเต็มให้มีเครื่องหมายบวกหรือลบ เช่น 123 จะแสดงเป็น +123
noshowpos	ยกเลิกการเซต showpos
showbase	ใช้ร่วมกับแฟล็ก oct หรือ hex เพื่อแสดงผลลัพธ์เป็นเลขฐานต่างๆ เช่น 1000 ₁₆ จะแสดงเป็น 0x1000
noshowbase	ยกเลิกการเซต showbase
boolalpha	แสดงค่าบูลีน ถ้าจริงเป็น true หรือค่าเท็จเป็น false
no boolalpha	ยกเลิกการเซต boolalpha
resetioflag(long f)	ใช้ยกเลิกการกำหนดรูปแบบการแสดงผล
setioflag(long f)	ใช้กำหนดรูปแบบการแสดงผล โดยต้องกำหนดแฟล็ก ถ้ามีมากกว่า 1 แฟล็ก ใช้โอเปอเรเตอร์บิตไวส์ออร์()เป็นตัวเชื่อม

การทดลองที่ 5_5 การใช้งานฟังก์ชันใน Library iomanip

1	#include <iostream>
2	#include <iomanip>
3	using namespace std;
4	int main()
5	{
6	float Raduis, Angle;
7	const int WIDTH = 9;
8	cout << "+" << setfill('=') << setw(44) << "+" << endl;
9	cout << ": Angle : Sine : Cosine : Tangent :" << endl;
10	cout << "+" << setfill('=') << setw(44) << "+" << endl;
11	cout << setfill(' ');
12	for (Angle = 0.0 ; Angle <= 360.0 ; Angle += 20) {
13	Raduis = (Angle > 0.0) ? (180.0f*3.14f)/Angle : 0.0;
14	cout << ":" << setw(WIDTH) << fixed << setprecision(2)
15	cout << Angle;
16	cout << ":" << setw(WIDTH) << setprecision(4) << sin(Raduis);


```
17         cout << " :." << setw(WIDTH) << cos(Raduis);
18         cout << " :." << setw(WIDTH) << tan(Raduis);
19         cout << " :." << endl;
20     }
21     cout << "+" << setfill('=') << setw(44) << "+" << endl;
22     return(0);
23 }
```

บันทึกผลการทดลอง

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slight shadow on the right side, suggesting it's resting on a surface.

การทดลองที่ 5 6 การใช้งานฟังก์ชันใน Library iomanip

```

1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4 int main()
5 {
6     cout << " Decimal : Octal : Hexa " << endl;
7     cout << setfill('.') << setw(30) << "-" << endl;
8     cout << setfill(' ') << showbase;
9     for(int Dec = 0 ; Dec <= 400; Dec += 50 )    {
10         cout << " " << right << setw(7) << setbase(10) << Dec << " : ";
11         cout << right << setw(7) << setbase(8) << Dec << " : ";
12         cout << left << setw(7) << setbase(16) << Dec << endl;
13     }
14     return(0);
15 }

```

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slight shadow on the right side, suggesting it's resting on a surface.

ภาษา C++ มาตรฐานได้ขยายไลบรารี C++ โดยเพิ่มคลาสสตริงเข้ามา แทนการใช้งานสตริงในรูปแบบอาร์เรย์ ซึ่งใช้เป็นชนิดข้อมูลแบบหนึ่ง ทำให้การใช้งานมีความสะดวกมากขึ้น และใช้การประกาศตัวแปรแบบเดียวกับชนิดข้อมูลเบื้องต้น แต่เนื่องจากเป็นคลาสในไลบรารี จึงต้องใช้กับเนมสเปส std และกำหนดไฟล์เฮดเดอร์ string เมื่อประกาศตัวแปรแล้วจะเรียกเป็นออบเจกต์ สามารถใช้งานในรูปแบบเดียวกับอาร์เรย์ตัวอักษร และสามารถใช้ประโยคคำสั่งรับค่า cin และแสดงผล cout ได้ และการเข้าถึงแต่ละตัวอักษรในรูปแบบเดียวกับอาร์เรย์

การเก็บค่าตัวอักษรในคลาสสตริง มีจำนวนมากเท่าไรก็ได้ ขึ้นกับว่ามีพื้นที่ว่างในหน่วยความจำเหลือจำนวนเท่าไร หากการประกาศไม่กำหนดค่าเริ่มต้น จะมีความยาวข้อความเป็น 0 เมื่อใดที่มีการเก็บข้อความจะเปลี่ยนขนาดตามจำนวนตัวอักษรที่รับเข้ามาให้อัตโนมัติ นอกจากนี้ยังใช้งานกับตัวดำเนินการได้ง่ายกว่า เมื่อมีการทำงานร่วมกันระหว่างตัวแปรคลาสสตริง

```
string S1; // S1 contains 0 characters
string S2 = "New York"; // S2 contains 8 characters
string S3 (60, '*'); // S3 contains 60 asterisks
string S4 = S3; // S4 contains 60 asterisks
string S5(S2, 4, 2); // S5 is the 2-character string "Yo"

char c = S2[0]; // assign 'N' to c

S2[3] = '-'; // changes S2 to "New-York"
```

การเก็บข้อความโดยใช้ตัวแปรคลาสสตริงสามารถใช้ตัวดำเนินการต่างๆได้เช่นเดียวกับชนิดข้อมูลเบื้องต้น ซึ่งได้ถูกสร้างให้สามารถใช้งานได้ในระบบแล้ว เช่น

```
Str3 = Str1 + Str2;
Str1 += Str2;
if (Str1 == Str3)
```

อย่างไรก็ตามจากคุณสมบัติของคลาสที่มีเมธอด ซึ่งเป็นฟังก์ชันของคลาสนั้น คลาสสตริงจึงมีเมธอดมาให้ใช้งาน ดังนี้

```
string S = "Microsoft Visual Studio.Net";
```

- **ฟังก์ชัน length()** เป็นฟังก์ชันการคืนค่าความยาวของข้อความที่เก็บ
ตัวอย่าง S.length() → 32
- **ฟังก์ชัน size()** เป็นฟังก์ชันการคืนค่าความยาวของข้อความที่เก็บ
ตัวอย่าง S.size() → 32
- **ฟังก์ชัน c_str()** เป็นฟังก์ชันการคืนค่าข้อความในแบบอาร์เรย์ของตัวอักษร
ตัวอย่าง S.c_str()
- **ฟังก์ชัน at()** เป็นฟังก์ชันการดึงค่าตัวอักษรที่เก็บอยู่ตามค่าตำแหน่งที่ระบุ
ตัวอย่าง S.at(7) → 'f'
- **ฟังก์ชัน append()** เป็นฟังก์ชันการเพิ่มข้อความสตริงลงในสตริงที่เรียกใช้
ตัวอย่าง S.append(" 2005")
- **ฟังก์ชัน compare()** เป็นฟังก์ชันการเปรียบเทียบข้อความที่เก็บอยู่ โดยคืนค่าเป็นตัวเลข 0 หมายถึงมีค่าเท่ากัน หากเป็น 1 หมายถึงมีค่ามากกว่า และเป็น -1 หมายถึงมีค่าน้อยกว่า
ตัวอย่าง S.compare(Name)
- **ฟังก์ชัน substr()** เป็นฟังก์ชันการสำเนาข้อความจากค่าตำแหน่งที่กำหนดไปตามจำนวนตัวอักษรที่ต้องการ
ตัวอย่าง S.substr(5, 4) → "soft"
- **ฟังก์ชัน find()** เป็นฟังก์ชันการค้นหาส่วนของข้อความในข้อความสตริงที่เก็บอยู่ โดยคืนค่าตำแหน่งที่พบครั้งแรกเป็นตัวเลขจำนวนเต็ม แต่หากไม่พบจะคืนค่าเป็น -1
ตัวอย่าง S.find("s") → 5
- **ฟังก์ชัน erase()** เป็นฟังก์ชันการลบตัวอักษรออกจากข้อความสตริงที่เก็บ ตามจำนวนตัวที่กำหนด
ตัวอย่าง S.erase(9, 7) → "Microsoft Studio.Net"

- ฟังก์ชัน `replace()` เป็นฟังก์ชันการแทนที่ข้อความลงในข้อความสตริงที่เก็บ ในค่าตำแหน่งที่กำหนด และลงในจำนวนที่ต้องการ
ตัวอย่าง `S.replace(10, 6, "C++")` → "Microsoft C++ Studio.Net"
- ฟังก์ชัน `clear()` เป็นฟังก์ชันการลบค่าข้อความที่เก็บทั้งหมดให้เป็น Empty String ("")
ตัวอย่าง `S.clear()` → ""
- ฟังก์ชัน `empty()` เป็นฟังก์ชันการตรวจสอบค่าข้อความว่าเป็น Empty String ("") หรือไม่ ถ้าใช่คืนค่าเป็น true ถ้าไม่ใช่คืนค่าเป็น false
ตัวอย่าง `S.empty()` → false
- ฟังก์ชัน `front()` เป็นฟังก์ชันคืนค่าอักขระตัวแรกของข้อความกลับมา
ตัวอย่าง `S.front()` → 'M'
- ฟังก์ชัน `back()` เป็นฟังก์ชันคืนค่าอักขระตัวสุดท้ายของข้อความกลับมา
ตัวอย่าง `S.back()` → 't'

การทดลองที่ 5_7 การใช้งานฟังก์ชันของคลาส string

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main()
5  {
6      string Str1 = "Microsoft";
7      string Str2 = Str1 + "Word";
8      string Str3;
9      // Display value and length
10     cout << "Value Str1 = " << Str1;
11     cout << ", Length Str1 = " << Str1.length() << endl;
12     cout << "Value Str2 = " << Str2;
13     cout << ", Length Str2 = " << Str2.length() << endl;
14     cout << "Value Str3 = " << Str3;
15     cout << ", Length Str3 = " << Str3.length() << "\n\n";
16     cout << "Enter text to Str3 : ";
17     cin >> Str3;
18     cout << "Now Length Str3 = " << Str3.length() << "\n\n";
19
20     switch( Str1.compare(Str2) )
21     {
22         case 0 :      cout << "Str1 equal Str2" << endl; break;
23         case 1 :      cout << "Str1 more than Str2" << endl; break;
24         case -1 :     cout << "Str1 less than Str2" << endl; break;

```

```

25     }
26     // find 's' from Str3
27     cout << "\nNow find 's' in Str3" << endl;
28     int pos = Str3.find("s");
29     if (pos > -1)
30         cout << "found 's' in Str3 at position " << pos << endl;
31     else
32         cout << "not found 's' in Str3." << endl;
33     // delete character in Str2
34     Str2.erase(5,4);
35     cout << "Aftet erase character to Str2 : " << Str2 << endl;
36
37     // replace character in Str1
38     Str1.replace(5,4,"chip");
39     cout << "Aftet replace character to Str1 : " << Str1 << endl;
40
41     // use substr in Str1 and Str2
42     cout << "\nString substr from Str1 and Str2 : ";
43     cout << Str1.substr(0,5) << Str2.substr(5,4) << endl;
44     return(0);
45 }

```

บันทึกผลการทดลองThis image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

2. ฟังก์ชันผู้ใช้กำหนด (User-Define Function)

ในภาษา C++ สามารถสร้างฟังก์ชันขึ้นมาใช้งานเองเพิ่มก็ฟังก์ชันก็ได้ตามความต้องการ โดยมีรูปแบบดังนี้

รูปแบบ

```
Function header → data-type Function Name (type var1,type var2,...)
{
Function body → variable declaration;
statements;
return(value);
}
```

ข้อกำหนดการใช้ฟังก์ชัน

- ฟังก์ชันที่สร้างจะประกอบด้วย 2 ส่วนคือ Function header และ Function body
- โดย Function header ประกอบด้วย data-type, a function name, a parameter declaration อยู่ภายใน ()
- โดย Function body ประกอบด้วยคำสั่งภาษา C++ อยู่ภายใน { }
- ใน Function body ต้องมีการใช้คำสั่ง return ที่บอกถึงจุดจบหรือเลิกการทำงาน และใช้คำสั่ง return เพื่อส่งค่าหนึ่งกลับจากการเรียกใช้ฟังก์ชัน
- สำหรับ Function header จะเริ่มด้วยการกำหนดประเภทของค่าที่ต้องการส่งกลับมาของฟังก์ชัน หากไม่กำหนดประเภทไว้จะถือว่าฟังก์ชันส่งค่ากลับเป็นตัวเลขจำนวนเต็ม(int) เสมอ แต่ถ้าไม่ต้องการให้ฟังก์ชันมีการส่งค่ากลับจะต้องใช้คำสั่ง void กำหนดไว้ เพื่อบอกว่าฟังก์ชันนี้ไม่ต้องการส่งค่ากลับ(ไม่มีการใช้คำสั่ง return)
- คำสั่ง return จะทำหน้าที่ 2 อย่าง คือ เป็นการบอกว่าจบการทำงานของฟังก์ชัน และ จะมีการส่งค่ากลับจากการเรียกใช้ โดย argument ของคำสั่ง return สามารถเป็น expression ได้

ตัวอย่าง

main()	/*	ส่งค่ากลับเป็น integer	*/
int Feet2Meter()	/*	ฟังก์ชันส่งค่ากลับชนิด integer	*/
void main()	/*	ไม่มีการส่งค่าใดๆ กลับไปที่ OS	*/
char findchar (str, ch)	/*	ส่งค่ากลับเป็นตัวอักษร 1 ตัว	*/

/*	User-define Function for calculate value of power by two */
int	square(int n)
{	
	return(n * n);
}	

2.1 การประกาศฟังก์ชัน (Function Prototype)

สำหรับในภาษา C++ นั้นฟังก์ชันที่สร้างขึ้น(user-define function)จะใช้งานได้ ต้องมีการประกาศฟังก์ชัน(function prototype)ให้รู้จักก่อน จึงสามารถเรียกใช้งานได้ ซึ่งมีรูปแบบการประกาศดังนี้

```
#include <...>
int FunctionName ( data-type );      /* function prototype */
void main()
{
    int v;
    V = FunctionName( variable );/* calling function */
}

int FunctionName( data-type variable )    /* Definition Function*/
{
    statements;
}
```

ตัวอย่าง

```
#include <iostream>
int power2( int );      /* function prototype */
int main()
{
    int x = 1;
    while ( x <= 10) {
        cout << power2(x) << "\n"; /* calling function */
        x = x + 1;
    }
    return(0);
}

int power2( int n)      /* Definition Function */
{
    return(n * n);
}
```

การทดลองที่ 5_8 การใช้งานฟังก์ชันที่สร้างเอง

```
1  #include <iostream>
2  using namespace std;
3  void CalCircle();
4
5  int main()
6  {
7      cout << "Program Calculate of Circle." << endl;
8      cout << "*****" << endl;
9      CalCircle();
10     return(0);
```

```
11 }  
12  
13 void CalCircle( )  
14 {  
15     float Radius, Area, Circumference;  
16     cout << "Input radius : ";  
17     cin >> Radius;  
18     Area = 3.14f*Radius*Radius;  
19     Circumference = 2*3.14f*Radius;;  
20     cout << endl;  
21     cout << "Area of circle : " << Area << endl;  
22     cout << "Circumference of circle : " << Circumference << endl;  
23     cout << "Diameter of circle : " << (2*Radius) << endl;  
24 }
```

บันทึกผลการทดลอง[illegible]

2.2 รูปแบบการส่งผ่านค่าระหว่างฟังก์ชัน

ในภาษาการเขียนโปรแกรมทั่วไปจะมีวิธีการส่งผ่านค่าข้อมูลให้กับฟังก์ชันที่ใช้งานได้ 2 แบบ คือ

1) Pass by Value

เป็นการส่งค่าของจากตัวแปร นิพจน์ หรือค่าคงที่ให้กับฟังก์ชันที่ถูกเรียกใช้งาน โดยค่าของจากตัวแปรที่ส่งให้นั้นไม่มีการเปลี่ยนแปลงเมื่อทำงานฟังก์ชันเสร็จแล้ว ดังนั้นจะมีเปลี่ยนแปลงค่าเฉพาะในฟังก์ชันเท่านั้น เรียกว่า เป็นแบบส่งค่าไปอย่างเดียว

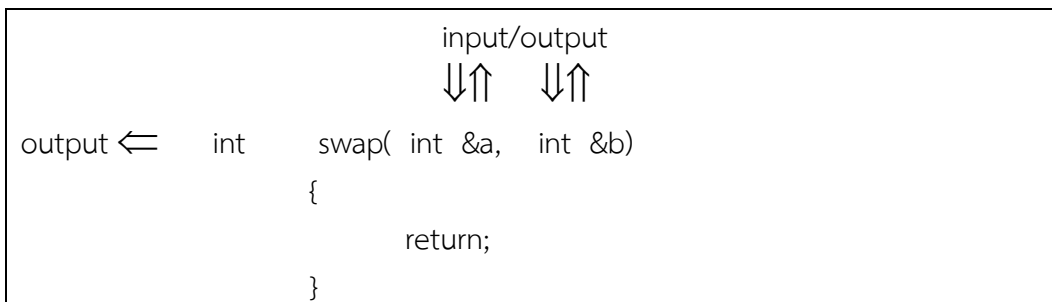
```

input
  ↓    ↓
output ⇐ int swap( int a, int b )
          {
            return;
          }

```


2) Pass by Reference

เป็นการส่งค่าของตัวแปรให้กับฟังก์ชันที่ถูกเรียกใช้งาน โดยค่าของตัวแปรที่ส่งให้มันจะเปลี่ยนแปลงไปตามการทำงานหลังจากฟังก์ชันทำงานเสร็จแล้ว เป็นแบบส่งค่าไปและรับค่ากลับมาด้วย



การทดลองที่ 5_9 การใช้งานฟังก์ชันด้วยการส่งค่าแบบ Pass by value

```

1  #include <iostream>
2  using namespace std;
3  int Min(int V1, int V2);
4  int Max(int V1, int V2);
5  int main()
6  {
7      int Value1, Value2;
8      cout << "Enter first number : ";
9      cin >> Value1;
10     cout << "Enter second number : ";
11     cin >> Value2;
12     cout << "Max value : " << Max(Value1, Value2) << endl;
13     cout << "Min value : " << Min(Value1, Value2) << endl;
14     return(0);
15 }
16
17 int Min(int V1, int V2)
18 {
19     if (V1 < V2) return(V1);
20     else return(V2);
21 }
22
23 int Max(int V1, int V2)
24 {
25     if (V1 > V2) return(V1);
26     else return(V2);
27 }

```

บันทึกผลการทดลอง

การทดลองที่ 5_10 การใช้งานฟังก์ชันด้วยการส่งค่าแบบ Pass by reference

```
1  #include <iostream>
2  using namespace std;
3  void Sort3(int &V1, int &V2, int &V3);
4  int main()
5  {
6      int Value1, Value2, Value3;
7      cout << "Enter three integer number : ";
8      cin >> Value1 >> Value2 >> Value3;
9      int Output1, Output2, Output3;
10     Output1 = Value1;
11     Output2 = Value2;
12     Output3 = Value3;
13     Sort3(Output1, Output2, Output3);
14     cout << Value1 << " " << Value2 << " " << Value3 ;
15     cout << " in sorted order is " ;
16     cout << Output1 << " " << Output2 << " " << Output3 << endl;
17     return(0);
18 }
19 void Sort3(int &V1, int &V2, int &V3)
20 {
21     int tmp;
22     if (V1 > V2) {
23         tmp = V1;
24         V1 = V2;
25         V2 = tmp;
26     }
27     if (V1 > V3) {
28         tmp = V1;
29         V1 = V3;
30         V3 = tmp;
31     }
32     if (V2 > V3) {
33         tmp = V2;
34         V2 = V3;
35         V3 = tmp;
36     }
37 }
```

บันทึกผลการทดลอง

การทดลองที่ 5_11 การใช้งานฟังก์ชันด้วยการส่งค่าแบบ Pass by reference

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  void ChangeString(string &S1, string &S2);
5  int main()
6  {
7      string Str1, Str2;
8
9      cout << "Enter first string : ";
10     cin >> Str1;
11     cout << "Enter second string : ";
12     cin >> Str2;
13     cout << "Data string before call function.\n";
14     cout << "Str1 = " << Str1 << endl;
15     cout << "Str2 = " << Str2 << endl;
16     ChangeString(Str1, Str2);
17     cout << "Data string after call function.\n";
18     cout << "Str1 = " << Str1 << endl;
19     cout << "Str2 = " << Str2 << endl;
20     ChangeString(Str1, Str2);
21     cout << "Data string after cal function.\n";
22     cout << "Str1 = " << Str1 << endl;
23     cout << "Str2 = " << Str2 << endl;
24     return(0);
25 }
26
27 void ChangeString(string &S1, string &S2)
28 {
29     string Temp;
30     Temp = S1;
```


}	
char func2 (char b) { int c, x; char name; }	

การหาค่ายกกำลัง จาก $y = x^n$	จาก $y = x^n = e^{(n \ln x)}$
float power(float x, int n) { float total=1; while (n > 0) { total *= x; n--; } return (total); }	float powern(float x, int n) { if (n < 1) return (1); else return (exp(n*log(x))); }

การทดลองที่ 5_12 การใช้งานฟังก์ชันกับขอบเขตของตัวแปร

1	#include <iostream>
2	using namespace std;
3	void a(void);
4	void b(void);
5	void c(void);
6	int x = 1;
7	int main()
8	{
9	int x = 5; /* local variable to main */
10	cout << "local x in outer scope of main is " << x << endl;
11	{
12	int x = 7;
13	cout << "local x in inner scope of main is " << x << endl;
14	}
15	cout << "local x in outer scope of main is " << x << endl ;
16	a(); /* a has automatic local x */
17	b(); /* b has static local x */
18	c(); /* c uses global x */
19	a(); /* a reinitialize automatic local x */
20	b(); /* static local x retains its previous value */
21	c(); /* global x also retains its value */
22	cout << "local x in main " << x << endl;
23	return(0);
24	}
25	
26	void a()

```

27 {
28     int    x = 25;          /* initialized each time is called */
29     cout << "\nlocal x is " << x << " on entering a()";
30     ++x;
31     cout << "\nlocal x is " << x << " before exiting a()\n";
32 }
33
34 void b()
35 {
36     static int    x = 50; /* static initializations only */
37     cout << "\nlocal static x is " << x << " on entering b()";
38     ++x;
39     cout << "\nlocal static x is " << x;
40     cout << " before exiting b()\n" << endl;
41 }
42 void c()
43 {
44     cout << "\nglobal x is " << x << " on entering c()";
45     x += 10;
46     cout << "\nglobal x is " << x << " before exiting c()\n";
47 }

```

บันทึกผลการทดลองThis image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

2.4 การสร้างฟังก์ชันแบบ Constant Parameters

เป็นการกำหนดค่าพารามิเตอร์ที่ส่งให้กับฟังก์ชันเป็นค่าคงที่ไม่สามารถแก้ไขค่าได้ โดยใช้คำว่า `const` ไว้ข้างหน้าชนิดข้อมูลของพารามิเตอร์ตัวนั้น

เช่น

```
void Example(const int a, int b, int c)
{
    b = a + 3;    // legal assignment
    a = c + 5;    //illegal assignment
}
```

การทดลองที่ 5_13 การใช้งานฟังก์ชันแบบ Constant Parameters

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  void ParseName(string &FirstName,string &LastName, const string FullName);
5
6  int main()
7  {
8      string Name = "Stroustrup, Bjarne";
9      string LastName, FirstName;
10
11     ParseName(FirstName,LastName,Name);
12     Name = FirstName + " " + LastName;
13     cout << "Name : " << Name << endl;
14     return(0);
15 }
16
17 void ParseName(string &FirstName,string &LastName, const string  FullName)
18 {
19     int I = FullName.find(",");
20     LastName = FullName.substr(0,I);
21     FirstName = FullName.substr(I+2,FullName.size());
22 }
```

บันทึกผลการทดลอง

2.5 การสร้างฟังก์ชันแบบ Default Parameters

เป็นการกำหนดค่าของพารามิเตอร์ให้กับฟังก์ชันโดยอัตโนมัติในกรณีที่มีการเรียกใช้ แบบไม่ได้กำหนดพารามิเตอร์ สำหรับวิธีการกำหนดค่า default ให้กับพารามิเตอร์จะทำเฉพาะส่วนการประกาศเท่านั้น เช่น

```
void f(int x = 7, int y = 3);    // prototype for f

f(5, 10); // legal call;
f(20);
f();
```

การทดลองที่ 5_14 การใช้งานฟังก์ชันแบบ Default Parameters

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int Summation(int StartNumber=1 ,int EndNumber=10);
5
6  int main()
7  {
8      int Start, End;
9      cout << "Enter start number : ";
10     cin >> Start;
11     cout << "Enter end number : ";
12     cin >> End;
13     cout << endl;
14     cout << "Summation of Number " << Start << " to " << End;
15     cout << " = " << Summation(Start,End) << endl;
16     cout << "Summation of Number " << 5 << " to " << 10;
17     cout << " = " << Summation(5) << endl;
18     cout << "Summation of Number " << 1 << " to " << 10;
19     cout << " = " << Summation() << endl;
20     return(0);
21 }
22
23 int Summation(int StartNumber, int EndNumber)
24 {
25     long Sum = 0;
26     for(int N = StartNumber ; N < EndNumber ; N++)
27         Sum += N;
28     return Sum;
29 }
```


บันทึกผลการทดลอง

2.6 การสร้างฟังก์ชันแบบ Overloading

ปกติการสร้างฟังก์ชันจะต้องมีชื่อฟังก์ชันที่ไม่ซ้ำกัน เหมือนการทำงานของตัวแปร แต่ในภาษา C++ สามารถให้ผู้สร้างฟังก์ชันที่มีชื่อฟังก์ชันเดียวกันมากกว่าหนึ่งได้ โดยมีเงื่อนไขคือมีจำนวนค่าพารามิเตอร์ไม่เท่ากัน หรือชนิดของพารามิเตอร์ต้องเป็นคนละชนิด เช่น

<pre>void Swap(float &x, float &y) { int tmp = x; x = y; y = tmp; } void Swap(int &x, int &y) { float tmp = x; x = y; y = tmp; }</pre>	<pre>float Area(float R) { return (R*R*3.14); } float Area(float L, float W) { return (L*W); }</pre>
---	---

การทดลองที่ 5_15 การใช้งานฟังก์ชันแบบ Overloading

1	<code>#include <iostream></code>
2	<code>#include <string></code>
3	<code>#include <iomanip></code>
4	<code>using namespace std;</code>
5	<code>// Define prototype function</code>
6	<code>void DisplayMenu();</code>
7	<code>float Area(const float Radius);</code>
8	<code>float Area(const float Length, const float Widht);</code>
9	<code>int main()</code>
10	<code>{</code>
11	<code> char Choice;</code>

```

12     bool Flag = true;
13     do {
14         DisplayMenu();
15         cin >> Choice;
16         if (Choice == '1') {
17             float Radius;
18             cout << "\nEnter radius : ";
19             cin >> Radius;
20             cout << "Area of Circle = " << fixed;
21             cout << setprecision(2) << Area(Radius) << endl;
22         }
23         else if (Choice == '2') {
24             float Length,Widht;
25             cout << "Enter length and width : ";
26             cin >> Length >> Widht;
27             cout << "Area of Rectangle = " << fixed;
28             cout << setprecision(2) << Area(Length,Widht);
29             cout << endl;
30         }
31         else if (Choice == '3') Flag = false;
32         else {
33             cout << "\nYou choose out of range is ";
34             cout << "not process.\n";
35         }
36     } while (Flag);
37     cout << "\n . . . Exit Program . . .\n";
38     return(0);
39 }
40
41 float Area(const float Radius)
42 {
43     return(3.14159F * Radius * Radius);
44 }
45 float Area(const float Length, const float Widht)
46 {
47     return(Length * Widht);
48 }
49
50 void DisplayMenu()
51 {
52     cout << endl;
53     cout << " Program Calculate Area " << endl;
54     cout << " 1. Circle" << endl;
55     cout << " 2. Rectangle" << endl;
56     cout << " 3. Exit" << endl;
57     cout << "Enter your choose number: ";
58 }

```

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

เป็นฟังก์ชันแบบเรียกตัวเอง โดยกำหนดให้ทำงานซ้ำๆ ตามจำนวนเงื่อนไขที่กำหนดไว้ เพื่อใช้การคำนวณที่ต้องการนำผลลัพธ์ที่ได้ก่อนหน้านี้มาใช้งาน และต้องขึ้นอยู่กับปัญหาที่นำไปใช้ด้วย

- 1) ต้องสามารถเขียนปัญหาให้อยู่ในรูปที่สามารถเรียกตัวเอง
- 2) ต้องมีเงื่อนไขสำหรับจบการทำงานของฟังก์ชัน

ใช้ Loop จาก $n! = 1 \times 2 \times 3 \times 4 \times 5 \times \dots \times n$
 จะได้ในรูปของผลคูณสะสม $fac = fac \times n$ โดย $n \geq 1$

```
long factorial(int n)
{
    long fac=1;
    for ( int x = 1; x <= n; x++ ) fac *= x;
    return (fac);
}
```

ใช้ Recursion Function

จาก $n! = 1 \times 2 \times 3 \times 4 \times 5 \times \dots \times n$

จะได้ $n! = n \times (n-1)!$ เงื่อนไข (1)

$(n-1)! = (n-1) \times (n-2)!$

:

$2! = 2 \times 1!$

$1! = 1 \times 0!$

$0! = 1$ เงื่อนไข (2)

```

long int factorial (int n)
{
    if ( n > 1 ) return(n * factorial(n-1));
    else return (1);
}

```

หากกำหนด fac = factorial (5) = 5!

Call	Variables (n)	factorial(n-1)
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

การทดลองที่ 5_16 การใช้งานฟังก์ชันแบบ Recursion function

```

1  #include <iostream>
2  using namespace std;
3  unsigned long Factorial1(int Number);
4  unsigned long Factorial2(int Number);
5  int main()
6  {
7      int Value;
8      cout << "Enter integer number(0-20) : ";
9      cin >> Value;
10     unsigned long Fac;
11     Fac = Factorial1(Value);
12     cout << "Factorial1 of " << Value << " is ";
13     cout << Fac << endl;
14     Fac = Factorial2(Value);
15     cout << "Factorial2 of " << Value << " is ";
16     cout << Fac << endl;
17     return(0);
18 }
19
20 // This function use loop
21 unsigned long Factorial1(int Number)
22 {
23     unsigned long Fac = 1;
24     for( int N = 1 ; N <= Number ; N++)
25         Fac *= N;
26     return(Fac);
27 }

```

```

28
29 // This function use recursion function
30 unsigned long Factorial2(int Number)
31 {
32     unsigned long Fac = 1;
33     if (Number > 1) return(Number * Factorial2(Number - 1));
34     else return(1);
35 }

```

บันทึกผลการทดลอง

การทดลองที่ 5_17 โปรแกรมการใช้งานฟังก์ชัน

```

1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  void UpperStr(char temp[80]);
5  int  MaxInt(int Val1, int Val2);
6  void Swap(int &a,int &b);
7  int main()
8  {
9      char Name[80];
10     int  N1, N2;
11     cout << "Enter two number(n1 n2): ";
12     cin >> N1 >> N2;
13     cout << "Enter string : ";
14     cin >> Name;
15     cout << endl;
16     cout << "\n Before call function...";
17     cout << "\n n1 = " << N1 << ", n2 = " << N2;
18     cout << ", string = " << Name;
19     cout << "\n Max value of n1 & n2 = " << MaxInt(N1,N2);
20     cout << "\n Press key to call function...";
21     getchar();
22     getchar();
23     swap(N1, N2);
24     UpperStr(Name);
25     cout << "\n After call function...";

```

```

26     cout << "\n n1 = " << N1 << ", n2 = " << N2;
27     cout << ", string = " << Name << endl;
28     return(0);
29 }
30
31 int MaxInt(int Val1, int Val2)
32 {
33     return ((Val1 > Val2)? Val1 : Val2) ;
34 }
35
36 void Swap(int &a,int &b)
37 {
38     int tmp;
39     tmp = a;      a = b;      b = tmp;
40 }
41
42 void UpperStr( char Str[80])
43 {
44     if (strlen(Str) > 0)
45         for( int index = 0 ; index < strlen(Str) ; index++)
46             Str[index] = toupper(Str[index]);
47 }

```

บันทึกผลการทดลองThe image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins or other markings on the paper.

ฟังก์ชันเทมเพลต

ฟังก์ชันเทมเพลตเป็นการสร้างฟังก์ชันเดียวที่สามารถรองรับการกำหนดพารามิเตอร์ที่มีชนิดข้อมูลได้ทุกประเภททำให้สะดวกต่อการใช้งาน และโปรแกรมที่เขียนมีประสิทธิภาพมากขึ้น แทนการสร้างฟังก์ชันแบบ Overloading ทำให้สามารถเขียนโปรแกรมแบบ Generic ได้

รูปแบบคำสั่ง

```
template<class identifier> function _declaration;  
template<typename identifier> function _declaration;
```

สามารถเขียนโดยใช้คำว่า class หรือ typename ก็ได้

เดิมหากต้องการให้ฟังก์ชันสามารถทำงานได้หลายชนิดข้อมูลก็ต้องเขียนเป็นฟังก์ชันแบบ Overloading เท่านั้น ซึ่งทำให้เกิดการสับสนเปลืองและยุ่งยาก

ดังตัวอย่างเป็นการสร้างฟังก์ชันหาผลรวมค่าสองจำนวนที่เป็นชนิด int , float, double

<pre>int sum(int a, int b) { int result; result = a + b; return(result); }</pre>
<pre>float sum(float a, float b) { float result; result = a + b; return(result); }</pre>
<pre>double sum(double a, double b) { double result; result = a + b; return(result); }</pre>

รูปแบบการเรียกใช้งาน

<pre>cout << sum(10, 20); cout << sum(10.0f, 20.0f); cout << sum(10.0, 20.0);</pre>
--

แต่หากเขียนในรูปฟังก์ชันเทมเพลต จะได้ดังนี้

```
Template<class T>
T sum( T a, T b)
{
    T result;
    result = a + b;
    return(result);
}
```

รูปแบบการเรียกใช้งาน

```
cout << sum<int>( 10, 20);
cout << sum<float>( 10.0f, 20.0f);
cout << sum<double>( 10.0, 20.0);
```

การทดลองที่ 5-18 การใช้งานฟังก์ชันเทมเพลต

```
1  #include<iostream>
2  using namespace std;
3  template <class T>
4  T Max( T a, T b);
5
6  int main()
7  {
8      int v1, v2;
9      cout << "Enter integer number.\n";
10     cin >> v1 >> v2;
11     cout << "Max = " << Max<int>(v1,v2) << endl;
12     double d1, d2;
13     cout << "\nEnter double number.\n";
14     cin >> d1 >> d2;
15     cout << "Max = " << Max<double>(d1,d2) << endl;
16     return(0);
17 }
18
19 template <class T>
20 T Max( T a, T b)
21 {
22     if (a > b)      return(a);
23     else return(b);
24 }
```


บันทึกผลการทดลอง

การทดลองที่ 5-19 การใช้งานฟังก์ชันเทมเพลต 2

```
1  #include<iostream>
2  using namespace std;
3
4  template <class T, class P>
5  T Average( P a[],int Max);
6  int main()
7  {
8      int intData[] = { 1,2,3,4,5,6,7,8,9,10 };
9      cout << "Average of data(integer) : ";
10     cout << Average<float, int >(intData, 10) << endl;
11     double doubleData[] = { 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0 };
12     cout << "Average of data(double) : ";
13     cout << Average<double, double>(doubleData, 10) << endl;
14     return(0);
15 }
16
17 template <class T, class P>
18 T Average( P a[], int Max)
19 {
20     T sum = 0.0;
21     for(int n = 0 ; n < Max ; n++)
22         sum += a[n];
23     return(sum / Max);
24 }
```

บันทึกผลการทดลอง

แบบฝึกหัดท้ายบท

1. จงเขียนโปรแกรมภาษา C++ สำหรับใช้คำนวณค่าตามสูตรที่กำหนดให้ โดยใช้ฟังก์ชันในไลบรารี cmath ในการทำงาน โดยโปรแกรมมีการรับค่า x ที่จะเอามาคำนวณตามสมการ

- $(\sin x)^2 \times (\cos x)^2$
- $e^{\frac{1}{2}\sqrt{\tan \cos x}}$
- $\frac{\left[\log \left(\frac{x^2}{1-x} \right) \right]}{\left[x^{5+x} \right]}$

2. จงเขียนโปรแกรมแสดงกราฟแท่งแนวนอน โดยโปรแกรมจะรับค่าตัวเลขจำนวน 5 ค่า แบบ Stream โดยการใส่ค่าจะใส่เพียงครั้งเดียวเท่านั้น ให้รับเป็นตัวเลขแบบจำนวนเต็มเท่านั้นห้ามรับเป็นข้อความ แล้วนำค่าที่ใส่มาแสดงเป็นเครื่องหมายดอกจันตามจำนวนค่านั้น โดยสร้างเป็นฟังก์ชันไว้สำหรับเรียกใช้ในการแสดงผลกราฟแท่งที่มีการรับค่าพารามิเตอร์หนึ่งค่า

ตัวอย่าง

```
Enter five number : 5 9 4 12 7
5 : *****
9 : *********
4 : ****
12 : *************
7 : *******
```

3. จงเขียนโปรแกรมตรวจสอบตัวเลขว่าเป็นตัวเลขจำนวนเฉพาะหรือไม่ โดยโปรแกรมจะทำการรับค่าตัวเลขจำนวนเต็ม และสร้างฟังก์ชันชื่อ CheckPrime สำหรับใช้ตรวจสอบค่าตัวเลขที่ส่งเข้ามา และมีการส่งค่าคืนกลับเป็น Boolean ที่ชื่อฟังก์ชัน โดยหากเป็นจำนวนเฉพาะคืนค่า true แต่หากไม่ใช่จำนวนเฉพาะคืนค่า false

4. จงเขียนโปรแกรมเพื่อคำนวณหาจำนวนเงินฝากธนาคาร โดยโปรแกรมให้รับข้อมูลดังนี้ จำนวนเงินต้น จำนวนปีที่ฝาก และอัตราดอกเบี้ย โดยให้สร้างฟังก์ชันทำงานแบบ Recursive เพื่อใช้ในการคำนวณเงินฝาก และโปรแกรมมีแสดงผลเงินฝากแต่ละปีที่ได้