

การเขียนโปรแกรมภาษา C++ WEEK #2

ดร.นิฐิตา เชิดชู โปรแกรมวิชาวิศวกรรมโทรคมนาคม
มหาวิทยาลัยราชภัฏนครปฐม

ตัวแปร (Variable) และชนิดของข้อมูล (Data Type)

□ ตัวแปร คือ สิ่งที่สามารถเปลี่ยนค่าไปได้เรื่อยๆ

□ การตั้งชื่อตัวแปร

ระวัง!!! ชื่อใน c++ นั้น case-sensitive
ดังนั้น Result ≠ result ≠ RESULT

□ ประกอบไปด้วย ตัวอักษร, ตัวเลข, หรือ _ เท่านั้น

□ ห้ามมีช่องว่างหรือเครื่องหมายอื่นๆ (เช่น . ,) อยู่ภายในชื่อที่ตั้งขึ้น

□ ชื่อตัวแปรต้องขึ้นต้นด้วยตัวอักษร (ในบางครั้งอาจยอมให้ขึ้นต้นด้วย _ ถ้าไม่ได้ถูกจองไว้เพื่อการใช้งานอย่างอื่นของคอมไพเลอร์แต่ละชนิด)

□ ห้ามขึ้นต้นด้วยตัวเลขเด็ดขาด

□ ชื่อที่ตั้งขึ้นต้องไม่ซ้ำกับคำที่ได้ถูกจองไว้ (reserved keywords)

```
asm, auto, bool, break, case, catch, char, class, const, const_cast, continue, default, delete,
do, double, dynamic_cast, else, enum, explicit, export, extern, false, float, for, friend, goto,
if, inline, int, long, mutable, namespace, new, operator, private, protected, public, register,
reinterpret_cast, return, short, signed, sizeof, static, static_cast, struct, switch, template,
this, throw, true, try, typedef, typeid, typename, union, unsigned, using, virtual, void,
volatile, wchar_t, while
```

```
and, and_eq, bitand, bitor, compl, not, not_eq, or, or_eq, xor, xor_eq
```

ชนิดของตัวแปร

- 1 byte เป็นหน่วยเล็กที่สุดของ memory ที่เราสามารถบริหารจัดการได้
- $8 \text{ bits} = 2^8 = 256 (0-255)$
- ข้อมูลพื้นฐานชนิดต่างๆ (*Size & Range จะมีความต่างกันในแต่ละคอมไพเลอร์)

Name	Description	Size*	Range*
char	Character or small integer.	1byte	signed: -128 to 127 unsigned: 0 to 255
short int (short)	Short Integer.	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	Long integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
bool	Boolean value. It can take one of two values: true or false.	1byte	true or false
float	Floating point number.	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	Double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	Long double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)
wchar_t	Wide character.	2 or 4 bytes	1 wide character

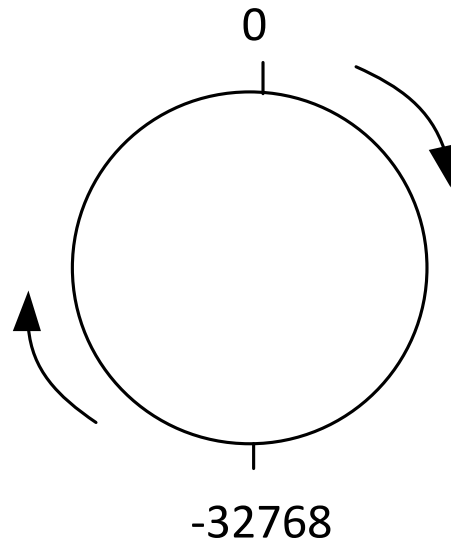
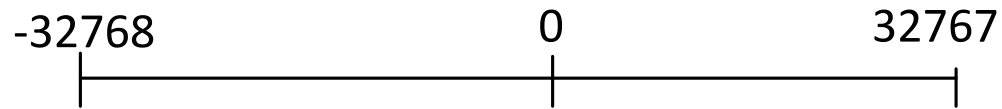
การประกาศตัวแปร

- ข้อมูลชนิดอื่นๆ ถ้าไม่ได้กำหนดว่าเป็น **signed** หรือ **unsigned** คอมไพเลอร์ส่วนใหญ่จะกำหนดให้เป็น **signed by default**
 - ยกเว้น **char** ถ้าเราต้องการเก็บข้อมูลที่เป็นตัวเลขให้กำหนดด้วยว่าเป็นแบบ **signed** หรือ **unsigned**
 - **short** และ **long** ถ้าใช้เดี่ยวๆ จะอนุมานว่าเป็น **short int** และ **long int** ตามลำดับ
- ```
short Year;
short int Year;
```
- **signed** และ **unsigned** ถ้าใช้เดี่ยวๆ จะอนุมานว่าเป็น **signed int** และ **unsigned int** ตามลำดับ

```
unsigned NextYear;
unsigned int NextYear;
```

# การกลับค่าของเลขจำนวนเต็ม

- การกำหนดชนิดของตัวแปรต้องกำหนดให้สอดคล้องกับช่วงของตัวเลขที่เราใช้งาน



# ตัวอย่าง

- จากโปรแกรมต่อไปนี้ ค่าของ **y** มีค่าเท่าไร

```
#include "stdafx.h"
#include <iostream>

using namespace std;

int main()
{
 short signed int a = 32767;
 short signed int y = a + 1;
 cout<<y;
 int x;
 cin>>x;
 return 0;
}
```



C:\Users\NPRU\Documents\Visual Studio 2010

-32768

# การตั้งชื่อตัวแปร

- ไม่ควรใช้ **reserve words** เช่น **int, bool, if, for** เป็นต้น

- ▣ **Reserved words in C**

```
auto const double float int short struct unsigned
break continue else for long signed switch void
case default enum goto register sizeof typedef volatile
char do extern if return static union while
```

- ▣ **Reserved words in C++**

```
asm dynamic_cast namespace reinterpret_cast try
bool explicit new static_cast typeid
catch false operator template typename
class friend private this using
const_cast inline public throw virtual
delete mutable protected true wchar_t
```

# การตั้งชื่อตัวแปร (ต่อ)

- ❑ ตัวแรกควรขึ้นต้นด้วยตัวอักษรหรือเครื่องหมาย **underscore** (**\_**)
- ❑ ชื่อตัวแปรสามารถประกอบไปด้วย ตัวอักษร ตัวเลข และ **underscore**
- ❑ ไม่ควรใส่อักขระพิเศษ เช่น **& % #** หรือ ช่องว่างในการตั้งชื่อตัวแปร
- ❑ ตัวอักษรเล็ก กับ ใหญ่มีความหมายต่างกัน เช่น
  - ❑ `int a = 10;`
  - ❑ `cout<< A; → error`  
A ไม่เท่ากับ a



# การประกาศตัวแปร

- ก่อนใช้งานตัวแปรใดๆ ต้องประกาศตัวแปรนั้นพร้อมทั้งชนิดของตัวแปร เช่น

- `int a;`

- `char c;`

- `float myNumber;`

- รูปแบบการประกาศตัวแปร

```
int a, b, c; =
int a;
int b;
int c;
```

- signed (-,0,+) vs unsigned (0,+)  
(signed by default for most compilers)

```
unsigned short int NumberOfSisters;
signed int MyAccountBalance;
```

# ตัวอย่าง: ตัวแปร

## □ การประกาศตัวแปร

<<Data type>> <<variable name>>;

int a;

## □ การใส่ค่าให้ตัวแปร

<<varname>>=<<value>>;

a=10;

## □ การนำตัวแปรไปใช้งาน

<<varname>>

a=a+1; //increments the value of a by 1

# ตัวอย่างโปรแกรม

```
// operating with variables

#include <iostream>
using namespace std;

int main ()
{
 // declaring variables:
 int a, b;
 int result;

 // process:
 a = 5;
 b = 2;
 a = a + 1;
 result = a - b;

 // print out the result:
 cout << result;

 // terminate the program:
 return 0;
}
```

4

# การใส่ค่าให้ตัวแปร (Variable Initialization)

□ ทำได้ 2 ชนิด

□ `type identifier = initial_value;`

■ `int a = 5;`

□ `Type identifier (initial_value);`

■ `int a(5);`

# ตัวอย่าง

```
// initialization of variables
```

```
#include <iostream>
using namespace std;
```

```
int main ()
```

```
{
```

```
 int a=5; // initial value = 5
```

```
 int b(2); // initial value = 2
```

```
 int result; // initial value
```

```
undetermined
```

```
 a = a + 3;
```

```
 result = a - b;
```

```
 cout << result;
```

```
 return 0;
```

```
}
```

6

# ขอบเขตของตัวแปร

```
#include <iostream>
using namespace std;
```

```
int Integer;
char aCharacter;
char string [20];
unsigned int NumberOfSons;
```

Global variables

```
int main ()
{
```

```
 unsigned short Age;
 float ANumber, AnotherOne;
```

Local variables

```
 cout << "Enter your age:";
 cin >> Age;
 ...
```

Instructions

```
}
```

# Strings

- ใช้เก็บตัวอักษรมากกว่า 1 ตัว
- ไม่ใช่ข้อมูลชนิดพื้นฐาน ทำให้ก่อนใช้งานต้องมีการ ใส่ไฟล์เฮดเดอร์

**#include <string>**

```
// my first string
#include <iostream>
#include <string>
using namespace std;

int main ()
{
 string mystring = "This is a string";
 cout << mystring;
 return 0;
}
```

This is a string

- การ **initialize strings**

```
string mystring = "This is a string";
string mystring ("This is a string");
```

# การใช้งาน strings

```
// my first string
#include <iostream>
#include <string>
using namespace std;

int main ()
{
 string mystring;
 mystring = "This is the initial string content";
 cout << mystring << endl;
 mystring = "This is a different string content";
 cout << mystring << endl;
 return 0;
}
```

```
This is the initial string content
This is a different string content
```



# ค่าคงที่ (Constant)

- `a = 5;`
- Constant
  - จำนวนเต็ม (Integer) เช่น
  - จำนวนทศนิยม (Float)
  - Character
  - Strings
  - boolean

# จำนวนเต็ม (Integer)

- จำนวนสิบ 75, 1776, 707, -273
- จำนวนแปด 0113
- จำนวนสิบหก 0x4b
- 75
- 75u
- 75l
- 75ul

# จำนวนทศนิยม (Floating)

- สามารถเป็นทศนิยม

- $e = 10^x$

```
3.14159 // 3.14159
6.02e23 // 6.02 x 10^23
1.6e-19 // 1.6 x 10^-19
3.0 // 3.0
```

# Character และ Strings

## □ Character

- ใช้เครื่องหมาย 'x' เพื่อแทนตัวอักษร x

```
'z'
'p'
"Hello world"
"How do you do?"
```

## □ Strings

- ใช้เครื่องหมาย "I am studying C++." เพื่อแทนประโยค

## □ การใช้ special characters

```
'\n'
'\t'
"Left \t Right"
"one\ntwo\nthree"
```

- การเขียนประโยคยาวๆ สามารถแบ่งเป็นหลายบรรทัดได้

```
"string expressed in \
two lines"
```

|    |                       |
|----|-----------------------|
| \n | newline               |
| \r | carriage return       |
| \t | tab                   |
| \v | vertical tab          |
| \b | backspace             |
| \f | form feed (page feed) |
| \a | alert (beep)          |
| \' | single quote (')      |
| \" | double quote (")      |
| \? | question mark (?)     |
| \\ | backslash (\)         |

# Boolean

□ **bool** (true หรือ false)

# Defined Constants (#define)

- ใช้เมื่อต้องการกำหนดค่าคงที่ของเราเอง เช่น
  - ▣ #define PI 3.414
  - ▣ #define NEWLINE '\n'
- ไม่ต้องมี ; ตอนจบ **statement** (ให้ นศ. ลองใส่ ;)

```
// defined constants: calculate circumference

#include <iostream>
using namespace std;

#define PI 3.14159
#define NEWLINE '\n'

int main ()
{
 double r=5.0; // radius
 double circle;

 circle = 2 * PI * r;
 cout << circle;
 cout << NEWLINE;

 return 0;
}
```

31.4159

# การกำหนดให้ข้อมูลเป็นชนิด Constant (const)

□ `const int x = 100;`

```
const int pathwidth = 100;
const char tabulator = '\t';
```

□ การใช้งานเหมือนกันกับ การประกาศ `int x = 100;` แต่

□ ค่าของ `x` จะไม่มีการเปลี่ยนแปลงหลังจากได้ **initialize** แล้ว



Operator



# ตัวดำเนินการ (OPERATORS)

- ตัวดำเนินการทางตัวเลข (Arithmetic):  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\%$
- ตัวดำเนินการเชิงเปรียบเทียบ (Relational):  $<$ ,  $>$ ,  $<=$ ,  $>=$ ,  $==$ ,  $!=$
- ตัวดำเนินการทางลอจิก (Logical):  $\&\&$ ,  $||$ ,  $!$
- ตัวดำเนินการแบบเงื่อนไข (Conditional):  $?$
- ตัวดำเนินการคอมม่า (Comma):  $,$
- ตัวดำเนินการทางบิต (Bitwise):  $\&$ ,  $|$ ,  $>>$ ,  $<<$
- การเพิ่ม-ลดค่า (Increment and Decrement):  $++$ ,  $--$
- การ **cast** ตัวแปร
- ตัวดำเนินการอื่นๆ

# ตัวดำเนินการทางตัวเลข (Arithmetic Operator)

□  $+, -, *, /, \%$

|   |                |
|---|----------------|
| + | addition       |
| - | subtraction    |
| * | multiplication |
| / | division       |
| % | modulo         |

□  $a = 11\%3;$

# ตัวดำเนินการเชิงเปรียบเทียบ (Relational Operator)

□ <, >, <=, >=, **==**, !=

|    |                          |
|----|--------------------------|
| == | Equal to                 |
| != | Not equal to             |
| >  | Greater than             |
| <  | Less than                |
| >= | Greater than or equal to |
| <= | Less than or equal to    |

ไม่เหมือนกับเครื่องหมายเท่ากับตัวเดียว

```
(7 == 5) // evaluates to false.
(5 > 4) // evaluates to true.
(3 != 2) // evaluates to true.
(6 >= 6) // evaluates to true.
(5 < 5) // evaluates to false.
```

```
(a == 5) // evaluates to false since a is not equal to 5.
(a*b >= c) // evaluates to true since (2*3 >= 6) is true.
(b+4 > a*c) // evaluates to false since (3+4 > 2*6) is false.
((b=2) == a) // evaluates to true.
```

# ตัวดำเนินการทางลอจิก (Logical Operator)

□ **&&, ||, !**

□ **!**

```
!(5 == 5) // evaluates to false because the expression at its right (5 == 5) is true.
!(6 <= 4) // evaluates to true because (6 <= 4) would be false.
!true // evaluates to false
!false // evaluates to true.
```

□ **&&**

| a     | b     | a && b |
|-------|-------|--------|
| true  | true  | true   |
| true  | false | false  |
| false | true  | false  |
| false | false | false  |

```
((5 == 5) && (3 > 6)) // evaluates to false (true && false).
((5 == 5) || (3 > 6)) // evaluates to true (true || false).
```

□ **||**

| a     | b     | a    b |
|-------|-------|--------|
| true  | true  | true   |
| true  | false | true   |
| false | true  | true   |
| false | false | false  |

# ตัวดำเนินการแบบเงื่อนไข (Conditional Operator)

- ใช้เมื่อต้องการทดสอบ **expression** ว่าเป็นจริงหรือไม่
- **Expression**
  - ▣ **condition ? result1 : result 2**
- ถ้า **condition** เป็นจริงจะ **return result 1**
- ถ้า **condition** เป็นเท็จจะ **return result 2**

```
7==5 ? 4 : 3 // returns 3, since 7 is not equal to 5.
7==5+2 ? 4 : 3 // returns 4, since 7 is equal to 5+2.
5>3 ? a : b // returns the value of a, since 5 is greater than 3.
a>b ? a : b // returns whichever is greater, a or b.
```

# ตัวอย่าง: การใช้งานตัวดำเนินการแบบเงื่อนไข

```
// conditional operator

#include <iostream>
using namespace std;

int main ()
{
 int a,b,c;

 a=2;
 b=7;
 c = (a>b) ? a : b;

 cout << c;

 return 0;
}
```

7

# ตัวดำเนินการคอมม่า (Comma Operator)

- ใช้เมื่อต้องการแยก **expression** หลายอันออกจากกัน ในขณะที่เราต้องการผลลัพธ์เพียง **expression** เดียว เช่น

```
a = (b=3, b+2);
```

- ▣  $b = 3;$
- ▣  $a = b + 2;$

# ตัวดำเนินการทางบิต (Bitwise Operator)

□ &, |, >>, <<

| operator | asm equivalent | description                      |
|----------|----------------|----------------------------------|
| &        | AND            | Bitwise AND                      |
|          | OR             | Bitwise Inclusive OR             |
| ^        | XOR            | Bitwise Exclusive OR             |
| ~        | NOT            | Unary complement (bit inversion) |
| <<       | SHL            | Shift Left                       |
| >>       | SHR            | Shift Right                      |



# การเพิ่ม-ลดค่า

- ++, --

- a++;

- ▣ a = a+1;

- a--;

- ▣ a = a-1;

# การเปลี่ยนชนิดของตัวแปรด้วยการ cast

- ใช้เปลี่ยนชนิดของตัวแปรชนิดหนึ่งให้เป็นอีกชนิดหนึ่ง
  - ▣ ด้วยการ (ชนิดของตัวแปรที่ต้องการเปลี่ยนเป็น) เช่น

```
int i;
float f = 3.14;
i = (int) f;
```

f = 3.14

f = 3

```
i = int (f);
```

# ตัวดำเนินการอื่นๆ

## □ sizeof()

```
a = sizeof (char);
```

## □ return ค่าเป็นจำนวน **byte** ที่ใช้เก็บข้อมูล

# การอินพุต เอาท์พุตเบื้องต้น

```
#include <iostream>
```

# Output

## □ Output (default: หน้าจอ)

### ▣ cout<<

```
cout << "Output sentence"; // prints Output sentence on screen
cout << 120; // prints number 120 on screen
cout << x; // prints the content of x on screen
```

### ▣ อะไรคือความแตกต่างระหว่าง

```
cout << "Hello"; // prints Hello
cout << Hello; // prints the content of Hello variable
```

### ▣ การผสมกันระหว่าง การพิมพ์ทั้งค่า **constant** และตัวแปร ออกทางหน้าจอ

```
cout << "Hello, I am " << age << " years old and my zipcode is " << zipcode;
```

# Output

- หากต้องการขึ้นบรรทัดใหม่หลังจากใช้ **cout** แล้วสามารถทำได้ ดังนี้

- **\n**

```
cout << "First sentence.\n ";
cout << "Second sentence.\nThird sentence.";
```

```
First sentence.
Second sentence.
Third sentence.
```

- **endl**

```
cout << "First sentence." << endl;
cout << "Second sentence." << endl;
```

```
First sentence.
Second sentence.
```

- **endl** จะทำการ **flush buffer** ทั้งหมด

# Input

## □ Input (default: แป้นพิมพ์)

### □ cin<<

```
int age;
cin >> age;
```

- การรับค่าต้องรอการกด **enter** จากแป้นพิมพ์เสมอ
- ชนิดของข้อมูลที่รับเข้ามาต้องสัมพันธ์กับชนิดของตัวแปรที่เรารอเก็บค่าด้วย
- รูปแบบต่างๆ ของการใช้ **cin**

```
cin >> a >> b;
```

```
cin >> a;
cin >> b;
```

ต้องใส่ input 2 ตัว โดยแยกกันด้วย **space** หรือ **enter** ก็ได้

# ตัวอย่าง: input

```
// i/o example

#include <iostream>
using namespace std;

int main ()
{
 int i;
 cout << "Please enter an integer value: ";
 cin >> i;
 cout << "The value you entered is " << i;
 cout << " and its double is " << i*2 << ".\n";
 return 0;
}
```

```
Please enter an integer value: 702
The value you entered is 702 and its double is
1404.
```



# cin กับ strings

- เนื่องจาก **cin** จะหยุดรับ **input** เมื่อมีการ **enter** หรือ **space** ระหว่างตัวอักษร
- ทำให้ถ้าข้อมูลที่ต้องการรับมาเป็น **string** ที่มี **space** อยู่ระหว่างคำ ไม่สามารถทำได้อย่างสะดวก
- ใช้ **getline()** แทน

# ตัวอย่าง cin กับ strings

```
// cin with strings
#include <iostream>
#include <string>
using namespace std;

int main ()
{
 string mystr;
 cout << "What's your name? ";
 getline (cin, mystr);
 cout << "Hello " << mystr << ".\n";
 cout << "What is your favorite team? ";
 getline (cin, mystr);
 cout << "I like " << mystr << " too!\n";
 return 0;
}
```

```
What's your name? Juan SouliÃÂÃÂ
Hello Juan SouliÃÂÃÂ.
What is your favorite team? The Isotopes
I like The Isotopes too!
```

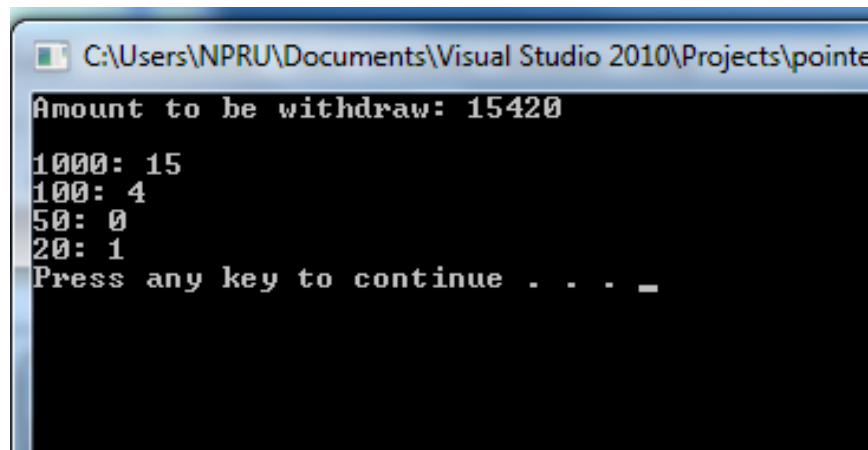
# Stringstreams

□ #include <sstream>

```
string mystr ("1204");
int myint;
stringstream(mystr) >> myint;
```

# การเขียนโปรแกรมการทำงานของ ATM

- ระบบถามว่าต้องการถอนเงินเท่าไร
- ผู้ใช้งานใส่จำนวนเงินที่ต้องการถอน (แบงค์เล็กที่สุด คือ แบงค์ 20)
- ระบบทำการคำนวณจำนวนแบงค์ทั้งหมดที่ต้องส่งให้ผู้ใช้งาน



```
C:\Users\NPRU\Documents\Visual Studio 2010\Projects\pointe
Amount to be withdraw: 15420
1000: 15
100: 4
50: 0
20: 1
Press any key to continue . . . _
```

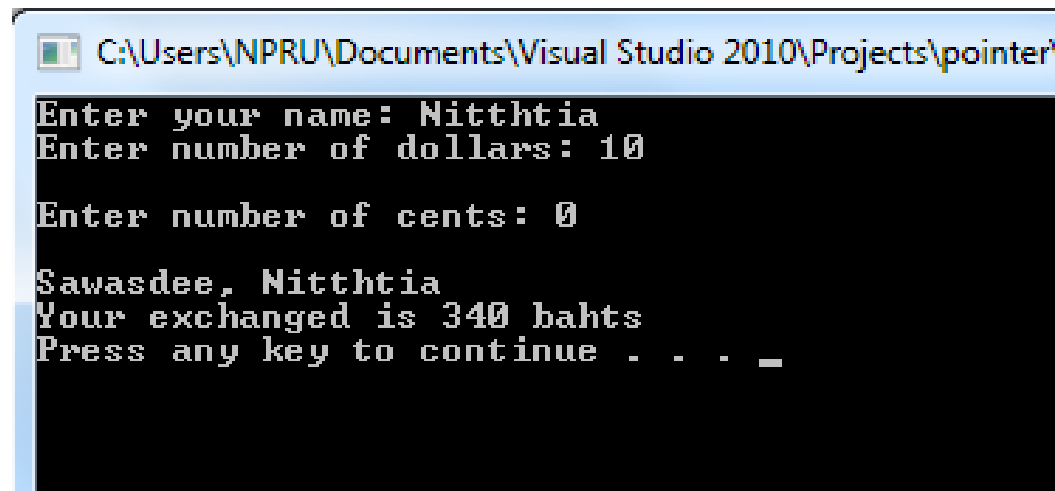
# เฉลย

```
int main()
{
 int withdraw;
 cout<<"Amount to be withdraw: ";
 cin>>withdraw;
 //calculate the number of bank note
 int thousand = withdraw/1000;
 int hundred = (withdraw - (thousand*1000))/100;
 int fifty = (withdraw - (thousand*1000) - (hundred * 100))/50;
 int twenty = (withdraw - (thousand*1000) - (hundred * 100) - (fifty * 50))/20;

 cout<<endl;
 cout<<"1000: "<<thousand<<endl;
 cout<<"100: "<<hundred<<endl;
 cout<<"50: "<<fifty<<endl;
 cout<<"20: "<<twenty<<endl;
 system("pause");
 return 0;
}
```

## แบบฝึกหัดที่ 2 โปรแกรมแลกเปลี่ยนจำนวนเงิน

- เพื่อแลกเปลี่ยนเงินสกุลดอลลาร์ ให้กลับมาเป็นเงินบาท
- อัตราแลกเปลี่ยน 1 **dollar** = 34 บาท



```
C:\Users\NPRU\Documents\Visual Studio 2010\Projects\pointer'\n\nEnter your name: Nitthtia\nEnter number of dollars: 10\n\nEnter number of cents: 0\n\nSawasdee, Nitthtia\nYour exchanged is 340 bahts\nPress any key to continue . . . _
```