

## Subroutine

Anan Phonphoem  
[anan@cpe.ku.ac.th](mailto:anan@cpe.ku.ac.th)

## Subroutine Calls

- Easy program by solving small problems
- 2 forms of subroutines:
  - Procedure (user-defined statements)
  - Function (user-defined operators)
- Must be defined in Declaration part

## Procedure

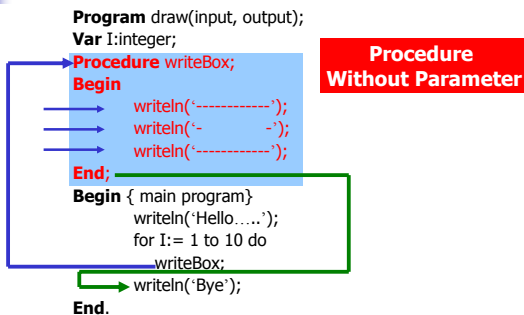
- Procedure is used to perform a task
- Examples of standard procedure:
  - Read( ); write( );
- Without parameter
  - Writeln;
- With parameter
  - Read(num); writeln('Hello');

## Procedure Structure

Same as a small Pascal program!

<b>Heading part</b>	Procedure drawLine;
<b>Declaration part</b>	Const maxCol = 10; Var column : integer;
<b>Statement part</b>	Begin for column := 1 to maxCol do write('*'); writeln; End;

## Procedure Operation



## Procedure

```
Program draw;
Var I:integer;
Procedure drawLine;
Const maxCol = 10;
Var column : integer;
Begin
  for column := 1 to maxCol do
    write('*');
  writeln;
End;
Begin {Main}
  for I:= 1 to 10 do
    drawLine;
End.
```

```
Program draw;
Var I:integer;
Procedure drawLineStar;
Const maxCol = 10;
Var column : integer;
Begin
  for column := 1 to maxCol do
    write('*');
  writeln;
End;
Procedure drawLineT;
Const maxCol = 10;
Var column : integer;
Begin
  for column := 1 to maxCol do
    write('T');
  writeln;
End;
Begin{Main}
  for I:= 1 to 10 do
    drawLineStar;
    drawLineT;
End.
```

```

Program draw;
Var I:integer;
Procedure drawLineStar;
Const maxCol = 10;
Var column : integer;
Begin
  for column := 1 to maxCol do
    write('*');
  writeln;
End;
Procedure drawLineT;
Const maxCol = 10;
Var column : integer;
Begin
  for column := 1 to maxCol do
    write('T');
  writeln;
End;
Begin {Main}
  for I:= 1 to 10 do
    drawLineStar;
    drawLineT;
  End.

```

```

Program drawnew;
Var I :integer;
  ch : char;
Procedure drawLine;
Const maxCol = 10;
Var column : integer;
Begin
  for column := 1 to maxCol do
    write(ch);
  writeln;
End;
Begin {Main}
  for I:= 1 to 10 do
    begin
      ch := '*';
      drawLine;
    end;
    ch := 'T';
    drawLine;
  End.

```

```

Program drawnew;
Var I :integer;
  ch : char;
Procedure drawLine;
Const maxCol = 10;
Var column : integer;
Begin
  for column := 1 to maxCol do
    write(ch);
  writeln;
End;
Begin {Main}
  for I:= 1 to 10 do
    begin
      ch := '*';
      drawLine;
    end;
    ch := 'T';
    drawLine;
  End.

```

```

Program drawAgain;
Var I :integer;
Procedure drawLine(ch:char);
Const maxCol = 10;
Var column : integer;
Begin
  for column := 1 to maxCol do
    write(ch);
  writeln;
End;
Begin {Main}
  for I:= 1 to 10 do
    drawLine('*');
    drawLine('T');
  End.

```

**Procedure With Parameter**

## Parameters

```

Program drawAgain;
Var I :integer;
  column : integer;
Procedure drawLine(ch:char);
Const maxCol = 10;
Var column : integer;
Begin
  for column := 1 to maxCol do
    write(ch);
  writeln(I); I:= 8;
End;
Begin
  column := 20; I := 3;
  drawLine('*');
  drawLine('T');
  writeln(column, I);
End.

```

Formal Parameter

Actual Parameter

## Procedure with parameter

- types of parameter
  - Value parameter [pass by value]
    - without **var**
    - bring-in value to the procedure
  - Variable parameter [pass by reference]
    - with **var**
    - bring-in and return the value

```

Program sample3(input,output);
Var num: integer;
Procedure read_input(var n:integer);
Begin
  write('Please input a number');
  readln(n);
End;
Procedure print_result(result: integer);
Begin
  writeln('-----');
  writeln('The result is',result );
  writeln('-----');
End;
Begin
  read_input(num);
  num := num *10;
  print_result(num);
End.

```

## Parameters

- Pass by **Value** Procedure
 

```

procedure Create1(A,B:integer)
procedure Create2(A:integer; B:integer);
procedure Create3(A:real; B:integer);

```
- Pass by **Reference** Procedure
 

```

procedure Draw1(var A,B:integer)
procedure Draw2(var A:integer; var B:integer);
procedure Draw3(var A:real; var B:integer);

```
- Mixed type
 

```

procedure DrawAgain1(var A:integer; B:integer);
procedure DrawAgain2(A,B:integer; var C:char);

```

```
Program sample3(input,output);
```

```
Var num: integer;
```

```
Procedure read_input(var n:integer);
```

```
Begin
```

```
write('Please input a number');
```

```
readln(n);
```

```
End;
```

```
Procedure print_result(result: integer);
```

```
Begin
```

```
writeln('-----');
```

```
writeln('The result is',result);
```

```
writeln('-----');
```

```
End;
```

```
Begin
```

```
read_input(num);
```

```
num := num *10;
```

```
print_result(num);
```

```
End.
```

## SCOPE RULE

```
1. num  
read_input  
print_result
```

```
2. n
```

```
3. result
```

## Scope Rule

Formal Parameter

```
Program drawAgain;
```

```
Var I :integer;
```

```
column : integer;
```

```
Procedure drawLine(ch:char);
```

```
Const maxCol = 10;
```

```
Var column : integer;
```

```
Begin
```

```
for column := 1 to maxCol do
```

```
write(ch);
```

```
writeln(1); I:= 8;
```

```
End;
```

```
Begin
```

```
column := 20; I := 3;
```

```
drawLine('*');
```

```
drawLine(' ');
```

```
writeln(column, I);
```

```
End.
```

Define the scope of the following variable:

- I
- Ch
- column

Actual Parameter