

Computer Science & Engineering 150A

Problem Solving Using Computers – Laboratory

Lecture 08 - Loop Statement

Shuai Xie

(Adapted from Derrick Stolee, Lin Liu & Shuai Xie)

Spring 2010

Announcement: CodeLab assignment

CSCE150A

Assignment

Review

While Loop

Basic
Operation

- The 4rd CodeLab assignment

Due: 23:59, Mar 17

If...else if...else...

CSCE150A

Assignment

Review

While Loop

Basic
Operation

```
if ( logical expression 1)
{
    executable statement 1;
    ...
}
else if ( logical expression 2)
{
    executable statement 2;
    ...
}
else
{
    executable statement 3;
    ...
}
```

- if...else if...else... is more efficient than if...if...if...

Multiple conditions in if statement

CSCE150A

Assignment

Review

While Loop

Basic
Operation

```
if ( ( logical expression 1) && ( logical  
expression 2) && ( logical expression 3 ) )  
{  
    executable statement;  
    ...  
}
```

Nested if statement

CSCE150A

Assignment

Review

While Loop

Basic
Operation

```
if ( logical expression )
{
    if ( logical expression )
    {
        executable statement;
        ...
    }
    else
    {
        executable statement;
        ...
    }
}
```

Switch statement

CSCE150A

Assignment

Review

While Loop

Basic
Operation

```
switch ( controlling_variable )
{
    case 1:
        executable group 1 ;
        break ;
    case 2:
        executable group 2 ;
        break ;
    ...

    default:
        executable group last ;
}
```

- controlling_variable: int, char
- char variable: 'c' ;
- break;
- other cases -- default :

- loop or iteration: do sth. more than once.
- C language supports three kinds of loops:
 - while loop
 - for loop
 - do while loop
- loop condition:
 - condition-controlled looping (while, do while)
A condition changed which will cause loop stop;
 - counter-controlled looping (for)
Loop runs for a certain number of times;
- Infinite loop: loop runs for ever.

While loop

CSCE150A

Assignment

Review

While Loop

Basic
Operation

```
while ( condition )  
{  
    executable1;  
    executable2;  
    change variable in conditional statement  
}
```

- Condition: any boolean value or expression
- Test condition. If it's true: execute loop body, repeats; false: skipped.
- Must make a change in conditional variable within the loop to avoid infinite loop.
- Get program to stop: "CTRL+C"

While loop: example 1

CSCE150A

Assignment

Review

While Loop

Basic
Operation

```
#include <stdio.h>

int main(void)
{
    int x = 0;
    while( x < 1 || x > 100 )
    {
        printf("Input a number between 1 and 100\n");
        scanf("%d",&x);
    }
    printf("The square of %d is %d\n", x, x * x );
}
```

Note: use scanf to change loop condition.

While loop: example 2

CSCE150A

Assignment

Review

While Loop

Basic
Operation

```
int sum(int n)
{
    int i = 1;
    int sum = 0;
    while ( i <= n )
    {
        sum = sum + i;
        i = i + 1;
    }
    return sum;
}
```

Note: use `i + 1` to change loop condition.

Exchange values of two variables x and y

CSCE150A

- Use the third temporary variable:

```
int tmp;
```

```
tmp = x;
```

```
x = y;
```

```
y = tmp;
```

Assignment

Review

While Loop

Basic
Operation

Order of three variables x, y and z

CSCE150A

- Six cases:

```
if ( x <= y && y <= z )
else if ( x <= z && z <= y )
else if ( y <= x && x <= z )
else if ( y <= z && z <= x )
else if ( z <= x && x <= y )
else if ( z <= y && y <= x )
```

Minimum value of three variables x, y and z

CSCE150A

- Three cases:

```
if ( x <= y && x <= z )
```

```
    min = x;
```

```
else if ( y <= x && y <= z )
```

```
    min = y;
```

```
else if ( z <= x && z <= y )
```

```
    min = z;
```

Assignment

Review

While Loop

Basic
Operation