

CISC 1600/1610 Computer Science I

Flow of control, Part 2 Loops

Professor Daniel Leeds
dleeds@fordham.edu
JMH 328A

Alternatives to “linear execution”

Repeated actions

```
> ./myProgram
Hello world.
Hello world.
Hello world.
Hello world.
>
```



2

The while loop

```
while ( condition )
    statement_to_repeat;
```

OR

```
while (condition)
{
    statement_to_repeat1;
    . . .
    statement_to_repeatN;
}
```

| block of
statements

3

condition – a Boolean expression

*Just a reminder from our earlier
if-else slides*

- Boolean expressions are either true or false
- Conditions often consist of **comparisons**
 - $\text{age} \geq 21$ // can buy drinks
 - $\text{age} < 4$ // can ride subway for free
 - $\text{year} = 2$ // you are a sophomore

4

How can we output “Hello world” 4 times?

```
int x=4;
while ( x>0 )
{
    cout << "Hello world.\n";
    x--;
}
```

Remember $x--$; same as $x=x-1$;
Repeats until $x \leq 0$

5

Execution of while loop

- If condition is true, enter while loop
 - Complete all statements in block
 - Return to top (re-evaluate condition)
- Otherwise, continue to statements beyond loop

6

Execution of while loop

- If condition is **true**, enter while loop
 - Complete all statements in block
 - Return to top (re-evaluate condition)
- Otherwise, continue to statements beyond loop

```
int x=2;
while ( x>0 )
{
    x--;
    cout << "Hello world.\n";
}
```

How many
"Hello world"s
are output?

7

What code will do this for us?

```
> ./myProgram
1 mississippi
2 mississippi
3 mississippi
4 mississippi
5 mississippi
>
```

8

a++ VS. ++a

- a++ returns value of a, then adds 1 to a
- ++a adds 1 to a, then returns value of a

Different results for:

```
int a=0;          int a=0;
while (a++ < 3)  while (++a < 3)
    cout << "Hi!\n";    cout << "Hi!\n";
```

9

Review of Friday program

```
Give me a number: 5
100/5 is 20
Give me a number: 33
100/33 is 3.333
Give me a number: 0
Cannot divide by 0!
```

```
cout << "Give num: ";
cin >> num;
while (num!=0) {
    cout<<"100/"<<num
    <<" is "<<100.0/num
    << endl;
    cout << "Give num: ";
    cin >> num;
}
```

do-while loop

- while evaluates condition, then performs statements if condition is **true**
- do-while performs statements, then evaluates condition to determine whether to perform statements again

```
do
{
    statement1;
    . . .
    statement N;
}
while ( condition );
```

11

Modification of Friday program

```
Give me a number: 5
100/5 is 20
Give me a number: 33
100/33 is 3.333
Give me a number: 0
Cannot divide by 0!
```

```
do {
    cout << "Give num: ";
    cin >> num;
    if(num!=0) {
        cout<<"100/"<<num
        <<" is "<<100.0/num
        << endl;
    }
} while(num!=0);
```

What does this code do?

```
int main () {
    int a=5;
    do {
        cout << "one ";
        a-=2;
        cout << "two\n";
    } while ( a > 0);
    return 0;
}
```

13

What does this code do?

```
int main () {
    int a=5;
    do {
        cout << "one ";
        a-=2;
        cout << "two\n";
    } while ( a != 0);
    return 0;
}
```

14

Beware infinite loops!

- Loops that never stop are called infinite loops
- Typically, write code so each loop will stop

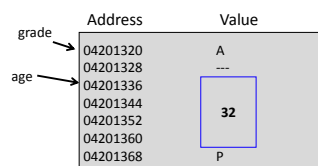
15

Example in scope:

```
int a=2, b=4;
while(a<10) {
    int b = a*5;
    cout << b << endl;
    a+=3;
}
cout << b << endl;
```

Variables – locations in memory

- Each variable indicates a location in memory
- Each location holds a value
- Value can change as program progresses
- Variable value exists before initialization



17

for loop
a while loop alternative

```
for ( init; condition; update )
{
    statement1;
    . . .
    statement N;
}
```

typical example:

```
int i;
for ( i=1; i<=5; i++)
{
    cout << i << " banana ";
}
```

18

`init` – initializes variable

`condition` – statement about variable,
must stay true for loop to keep running

`update` – updates the variable after each
loop execution

19

Reviewing scope

Counter `i` exists outside of loop

```
int i;
for ( i=1; i<=5; i++)
{
    cout << i << " banana ";
}
```

Counter `i` exists **only** inside of loop

```
for ( int i=1; i<=5; i++)
{
    cout << i << " banana ";
}
```

20

What does this code do?

```
int main () {
    int i, product=1;
    for ( i=1; i<=5; i++);
        product = product*i;
    cout << i << "! = " << product << endl;

    return 0;
}
```

21

Beware the misplaced ;

Placing a semicolon after the parentheses of a `for` loop causes an empty statement as the body of the loop

22

Picking a loop

- `do-while` if you need to perform the action at least once
- `for` if there is a standard repeated mathematical update to your loop variable (e.g., `count++`)
- `while` loop for less-standard loop variable updates

“loop variable” is the variable tested by the condition in your given loop

23