

# Language Basics – Flow of Control

---



**Kate Gregory**

@gregcons [www.gregcons.com/kateblog](http://www.gregcons.com/kateblog)



# Flow of Control

```
int i1 = 1;  
  
std::cout << "i1= "  
    << i1 << std::endl;  
  
int i2;  
  
i2 = 2;  
  
std::cout << "i2= "  
    << i2 << std::endl;
```

◀ Normally from one line to the next



# Standard Constructs in Almost Every Language

**if**

**while**

**Calling a function**

**return**



# Different in C++

for

break

continue

switch

Immediate if

goto



```
if (x>3) y=7;
```

```
if (x>3)  
    y=7;
```

```
if (x>3)  
{  
    y=7;  
}
```

```
else
```

```
{  
    x++;  
}
```

|f

◀ if (condition) action

◀ Round brackets are not optional

◀ When “action” is multiple lines,  
use braces

◀ else is optional



# While

`while (condition) action`

Check is before first time

Same rules as if – round  
brackets, braces

Can exit early using break



# For

```
for (int i = 0; i < 10; i++)  
{  
    cout << i << endl;  
}
```

```
for (int i = 0;  
     i < 10;  
     i++)
```

- ◀ Initializer
- ◀ Condition
- ◀ Loop expression



# Switch

**Many if  
statements**

**Less nesting**

**Must be integral  
type or enum**





# Switch

```
switch(expression)
{
case value1:
    // ...action...
    break;
default:
    // ...action...
}
```



# Switch

```
switch(expression)
{
case value1:
    // ...action...
    break;
case value2:
case value3:
    // ...action...
    break;
default:
    // ...action...
}
```



# Switch

```
int x = someFunction(stuff);  
switch(x)  
{  
case value1:  
    // ...action...  
    break;  
default:  
    // ...action...  
}
```



# Switch

```
Thing t = someFunction(stuff);  
switch(t.getStatus())  
{  
  case value1:  
    // ...action...  
    break;  
  default:  
    // ...action...  
}
```



# Switch

```
switch(Thing t = someFunction(stuff); t.getStatus())  
{  
  case value1:  
    // ...action...  
    break;  
  default:  
    // ...action...  
}
```



# Immediate If

```
if (something)  
    result = 7;  
else  
    result = 302;
```

```
result = something? 7 : 302;
```



# Summary



C++ has the usual flow-of-control constructs plus some you many not have in other languages

Use according to idiom

**for/while:**

- Do you know the number of iterations in advance?

**switch/series of if**

- 3 or more use a switch

**Immediate if**

- Are all three parts short enough to fit on one line?