

Syntax for C++ Control Structures — Some Examples

If Statement

General form of an if without an else:

```
if (a boolean expression goes here)  
{  
    a sequence of statements goes here  
}
```

- Example 1

```
if (c_p < p_len)  
{  
    c_p++;  
    output << t[c_p] << '\n';  
}
```

- Example 2

```
if (c_p < p_len)  
{  
    c_p++;  
    if (t[c_p] == 'a')  
    {  
        count++;  
    }  
}
```

General form of an if with an else:

```
if (a boolean expression goes here)  
{  
    a sequence of statements goes here  
}  
else  
{  
    a sequence of statements goes here  
}
```

- Example 1

```
if (c_p < p_len)  
{  
    c_p++;  
    output << t[c_p] << '\n';  
}  
else  
{  
    output << "All done!";  
}
```

- Example 2

```
if (('A' <= letter) and (letter <= 'Z'))  
{
```

```

    output << "It's an upper case letter\n";
}
else
{
    output << "It's not an upper case letter.\n";
}

```

General form of an if with an else if:

```

if (a boolean expression goes here)
{
    a sequence of statements goes here
}
else if (a boolean expression goes here)
{
    a sequence of statements goes here
}
else if (a boolean expression goes here)
{
    a sequence of statements goes here
}
...
else
{
    a sequence of statements goes here
}

```

- Example 1

```

if (('A' <= letter) and (letter <= 'Z'))
{
    output << "It's an upper case letter\n";
}
else if (('a' <= letter) and (letter <= 'z'))
{
    output << "It's a lower case letter.\n";
}
else if (('0' <= letter) and (letter <= '9'))
{
    output << "It's a digit letter.\n";
}
else
{
    output << "Garbage in, garbage out!\n";
}

```

- Example 2

```

if (button_pushed == RECTANGLE)
{
    output << "Enter length: ";
    input >> length;
    output << "Enter height: ";
    input >> height;
    output << "Area is " << length * height << '\n';
}

```

```
else if (button_pushed == CYLINDER)
{
  output << "Enter raduis: ";
  input >> radius;
  output << "Enter height: ";
  input >> height;
  ouput << "Area is " << height * 2 * pi * raduis << '\n';
}
else if (button_pushed == RIGHT_TRIANGLE)
{
  output << "Enter base: ";
  input >> radius;
  output << "Enter height: ";
  input >> height;
  ouput << "Area is " << height * base * 0.5 << '\n';
}
else
{
  output << "Sorry - try again!";
}
```

While Statement

General form of a while statement:

```
while (a boolean expression goes here)  
{  
    a sequence of statements goes here  
}
```

- Example 1

```
while (item_num < num_items)  
{  
    input >> item;  
    sum_so_far = item * item_num + sum_so_far;  
    item_num++;  
}
```

- Example 2

```
while (big != small)  
{  
    big = big - small;  
    if (big < small)  
    {  
        big &= small;  
    }  
}
```

- Example 3

```
while (big != small)  
{  
    while (big > small)  
    {  
        big = big - small;  
    }  
    big &= small;  
}
```

Case_select Statement

General form of a case_select statement:

```
case_select (an integer or character object goes here)
{
  case an integer or character expression goes here:
  {
    a sequence of statements goes here
  }
  break;
  case an integer or character expression goes here:
  {
    a sequence of statements goes here
  }
  break;
  ...
  default :
  {
    a sequence of statements goes here
  }
  break;
}
```

- Example 1

```
case_select (button_pushed)
{
  case RECTANGLE:
  {
    output << "Enter length: ";
    input >> length;
    output << "Enter height: ";
    input >> height;
    output << "Area is " << length * height << '\n';
  }
  break;
  case CYLINDER:
  {
    output << "Enter radius: ";
    input >> radius;
    output << "Enter height: ";
    input >> height;
    output << "Area is " << height * 2 * pi * radius << '\n';
  }
  break;
}
```

```

case RIGHT_TRIANGLE:
{
    output << "Enter base: ";
    input >> radius;
    output << "Enter height: ";
    input >> height;
    ouput << "Area is " << height * base * 0.5 << '\n';
}
break;
default:
{
    output << "Sorry - try again!";
}
break;
}

```

- Example 2

```

case_select (character_entered)
{
    case 'a':
    {
        output << "Enter an item: ";
        input >> item;
        s.Add (item);
    }
    break;
    case 'r' :
    {
        output << "Enter an item: ";
        input >> item;
        s.Remove (item);
    }
    break;
    case 's' :
    {
        s1 &= s2;
    }
    break;
    default:
    {
        output << "Sorry - make another selection!";
    }
    break;
}

```