

# C++ Notes

Jonathan Bowden

4th April 2018

## 1 Operators

- Variable Address

```
cout << &A << endl
```

- Joining Strings

```
string combinedStrings = x + ' ' + y; // just use plus to  
    combine string
```

- Incrementing

```
cout << "INCREMENTATION EXAMPLES" << endl;  
int d = 1;  
cout << d++ << endl; // returns one because it outputs d first,  
    then increments (POST-INCREMENTATION)  
cout << ++d << endl; // increments first (PRE-INCREMENTATION)
```

```
cout << "DECREMENTATION EXAMPLES" << endl;  
int e = 1;  
cout << e-- << endl; // returns one because it outputs d first,  
    then decrements (POST-DECREMENTATION)  
cout << --e << endl; // decrements first (PRE-DECREMENTATION)
```

- And (conjunction)

```
cout << ((7 < 5) && (5 != 10)) << endl;
```

- OR (disjunction)

```
cout << ((7 < 5) || (5 != 10)) << endl;
```

- Bitwise Operators

```
/*  
Bitwise AND - &  
Bitwise OR - |  
Bitwise NOT - ~ (tilde)  
Bitwise XOR - ^ (caret)  
Bitwise left shift - <<  
Bitwise right shift - >>  
*/
```

## 2 Logic

- IF-THEN-ELSE

```
if (a > b) {
    cout << a << " > " << b << endl;
}
else if (a < b)
    cout << a << " < " << b << endl;
else
    cout << "conditions not met" << endl;
```

- SWITCH-CASE Need to remember the break; command at the end of each case, or C++ will execute sequentially.

```
int x = 0;

switch (x) //executes all code after condition is met
{
    case 0:
        cout << "code here when case is 0" << endl;
        break;
    case 25:
        cout << "code here when case is 25" << endl;
        break;
    case 50:
        cout << "code here when case is 50" << endl;
        break; // stops the rest of the switch code being executed
    default:
        cout << "code here when value is nothing else" << endl;
}
```

- CONDITIONAL OPERATOR - ?

```
string message = (a > b) ? "a > b" : "a <= b";

cout << ((a > b ? a : b)) + 10 << endl; // add 10 to the higher
number
```

- For Loops

```
for (init; condition; inc/dec)

for (int i = 0; i < 5; i++)
{
    cout << "HELLO" << endl;
}

int arr[100];

for (int i = 0; i < 100; i++)
{
    arr[i] = i;
    cout << arr[i] << endl;
}
```

- Do Loops

```

while (--i) // putting the increment before the variable, checks
            the "next" condition before executing loop
{
    cout << i << endl;
}

int arr[sizeofarray];

while(i < sizeofarray)
{
    arr[i] = 10 * i;
    cout << arr[i++] << endl; // First send to the ouput, then
                                increment
}

do
{
    cout << "lala";
} while (i); //check condition at end

```

- Continue and Next

Continue keeps the loop going, break does not.

```

for (int i = 1; i <= 10; i++) // i = 2
{
    //if (i == 5)
    //    continue; // everything after the continue won't
                    be executed, but the loop won't be stopped.

    //if (i == 5)
    //    break; // everything after break won't be executed
                and the loop is stopped.

    for (int j = 1; j <= 10; j++) // j = 1
    {
        if (j == 5)
            break; //exits the loop, continue just
                    skips the 5th one
        cout.width(4);
        cout << i * j;
    }

    cout << endl;
}

for (int i = 1, j = 1; i <= 10; i++)
{

```

```

        cout.width(4);
        cout << i * j;

        if (i == 10)
        {
            j++;
            i = 0;
            cout << endl;
        }

        if (j == 10 + 1) //add plus one to see the 10th row
            break;
    }

```

### 3 Variables

- Arrays

```

int arr[4];
arr[0] = 10;

int biarr[3][4] = { 0 };
int triarr[2][3][2];

// the first array item represents the address of the entire array
// as well:
cout << "Array address: " << &arr << endl;

```

### 4 Code snippets

- Data Validation

```

bool isValid(string error_msg)
{
    if (cin.rdstate()) // state is wrong when not equal to zero
    {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        system("cls");
        initMenu();
        cout << error_msg << endl;
        return false; // return leaves the function.
    }
    return true;
}

// This is used in the main code as follows:

do { cin >> a; } while (!isValid("The input is invalid"));
areaSquare(a);

// Can also use an overload function without an error message:

```

```

bool isValid()
{
    if (cin.rdstate()) // state is wrong when not equal to zero
    {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        system("cls");
        initMenu();
        return false; // return leaves the function.
    }
    return true;
}

// Used as follows:

do { cout << "Enter the radius:" << endl; cin >> r; } while (!
    isValid());
areaCircle(r);

```