# C++ Reference Guide

Saddleback College

## PREPROCESSOR DIRECTIVES

**Header Files (syntax)**

```
#include <headerfile>
```

**Common Headers**

```
iostream, iomanip
```

**Namespace**

```
using namespace std;
```

## DECLARATION SECTION

**Data Types**

```
int, float, char, bool
```

**c-string syntax:**

```
char varName [size]
```

**Variable Declaration Syntax**

```
datatype varName ;
```

**Constant Declaration Syntax**

```
const datatype CONST_NAME = value ;
```

**Data Table**

```
- Doc to the right of variables
- Doc above constants
// use of variable - IN , OUT or CALC
```

## ARITHMETIC OPERATORS (in order of precedence)

() Parenthesis
* Multiplication, / Division, % Modulus
+ Addition, - Subtraction

**Note: Modulus is only valid with int**

## RELATIONAL OPERATORS

| | | | |
|---|---|---|---|
| < | Less Than | <= | Less Than or Equal To |
| > | Greater Than | >= | Greater Than or Equal To |
| == | Equal To | != | Not Equal to |

## LOGICAL OPERATORS

&& Logical AND
|| Logical OR
! Logical NOT

## INPUT/OUTPUT OPERATIONS

**INPUT**

>> Extraction Operator (for int or float)

```
cin >> variable ;
```

.getline (for c-strings)

```
cin.getline(variable, size);
```

.get (for single characters)

```
cin .get(variable );
```

.ignore (to flush the input buffer)

```
cin .ignore( #_of_chars, char_value );
Example: cin.ignore( 10000, '\n');
```

**OUTPUT**

<< Insertion Operator

```
cout << var_const_or_expr << var_const_or_expr ;
Example: cout << "Name: " << userName;
```

**endl** - goes to a new line

**setw(*int_value*)** - specifies output field

**FORMATTING FLOATS**

**fixed** - fixed floating pt notation
**setprecision(*int_value*)**
**showpoint**

## SELECTION STATEMENTS

**IF-THEN (one-way)**

```
if ( boolean expression )
{
    true statements ;
}
```

**IF-THEN-ELSE (two-way)**

```
if ( boolean expression )
{
    true statements ;
}
else
{
    false statements ;
}
```

**SWITCH (multi-way)**

```
switch (expression )
{
case constant1 :   statement(s) ;
                   break;
case constant2 :   statement(s) ;
                   break;
   :
case constant(n) : statement(s) ;
                   break;
default :          statement(s) ;
                   break;
}
```

**NOTE: default is optional**

## REPETITION STATEMENTS

**FOR LOOP (known # of iterations)**

```
for ( initialize ; check ; change )
{
    loop statements;
}
```

**WHILE LOOP (unknown # of iterations)**

```
initialize statement;
while ( boolean expression )
{
    loop statements;
    Change statement
}
```

**DO-WHILE LOOP (post-test loop)**

```
do
{
    initialize/change statement;
    loop statements;
} while ( boolean expression );
```