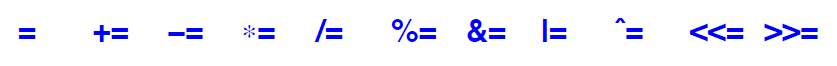
* **Introduction:** In C++, an *assignment* is an expression, a *function call* is an expression, the *construction of an object* is an expression, etc.
* The following assignment operators are possible in C++.



* **isspace(c):** Provides standard test for whitespace. Returns a non-zero value if c is a whitespace character and 0 otherwise.
* **isdigit(c):** Tests if a character is a digit.
* **isalpha(c):** Tests if a character is an alphabet.
* **isalnum(c):** Test if a character is a digit or a letter.
* These test functions are available in #include<cctype>
* **Constant Expressions –**
* There are a variety of reasons why someone might want a named constant rather than a literal or a value stored in a variable –
* Named constants make the code easier to read, understand and maintain.
* A variable might be changed. So, we have to be more careful in our reasoning than a constant.
* The language requires constant expressions for array sizes, case labels and template value arguments.
* Embedded-system programmers like to put immutable data into read-only memory. This is because read-only memory is cheaper than dynamic memory (in terms of cost and energy consumption) and often more plentiful. Also, data in read-only memory is immune to most system crashes.
* If initialisation is done at compile time, there can be no data races in that object in a multi-threaded system.
* Sometimes, evaluating something once at compile time gives significantly better performance than doing so a million times at runtime.