**Programming by contract**

**Assertion**: A logical proposition that can be true or false.

**Precondition**: Assertions that must be true on entry into an operation or function for the post condition to be guaranteed.

**Postconditions**: Assertions that state what results are expected at the exit of an operation or function, assuming that the preconditions are true.

**Invariant**: Aspects of the state that remain unchanged by the invocation of a method.

Every feature, or method, starts with a precondition that must be satisfied by the consumer of the routine. Each feature ends with postconditions which the supplier guarantees to be true.

What are the assertions, preconditions, postconditions and the invariants for this function?

*Example 1:*

//The function summation adds 10 to part

**//The precondition** part is not INT\_MAX-10

**//Post condition:** The value of sum > Part

Int summation (int part)

{

int sum;

sum=part+10;

return sum;

}

What are the preconditions and postconditions for the following?

*Example 2:*

//deposit money into bank account

double depositMoney( double amount)

*Example 3:*

//Remove an item from a list

list \* remove(list \*lst)

*Example 4:*

//divide dividend by divisor and return the result

int divide(int dividend, int divisor)

*Example 5:*

//return true if vowel

bool isVowel(char character)

What are the invariants?

*Example 1:*

boolean search(int term, int array[])

*Example 2:*i=0;

int k = 9;

for(; i<10; i++)

k--;

**assert function**

The assert function will terminate the program (usually with a message) if its argument turns out to be false. It is commonly used during debugging to make the program fail more obviously if an unexpected condition occurs.

assert(length >= 0);

assert(length >= 0 && "length can't be negative!");

assert("length can't be negative!" && length >= 0);

assert(("length can't be negative!", length >= 0));

//commenting the next line would enable the call to assert

//#define NDEBUG

#include <assert.h>

#include <iostream>

using namespace std;

int main(){

int length;

cin>>length;

//assert(("length is wrong", length>=0));

//assert(length>=0 && "what is going on");

assert("length is wrong" && length>=0);

}