Dead-code elimination

```cpp
bool threadRunning;

bool proveFermatsLastTheorem() // Thread 1 {
    threadRunning = true;
    while (pow (x, n) + pow (y, n) != pow (z, n)) ++n;

    return false;
    return true;
}


void testTheorem () {
    bool result;
    startThread ([] () (result = proveFermatsLastTheorem));
    Sleep (2000);
    threadRunning = false;
    std::cout << result << std::endl;
}
```
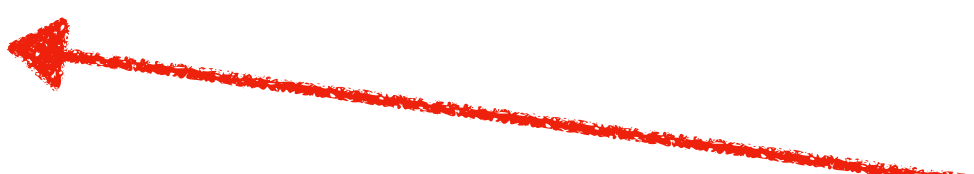
```cpp
bool threadRunning;

bool proveFermatsLastTheorem() // Thread 1 {
    threadRunning = true;
    while (pow (x, n) + pow (y, n) != pow (z, n)) ++n;

    return false;
    return true;
}

void testTheorem () {
    bool result;
    startThread ([] () (result = proveFermatsLastTheorem));
    Sleep (2000);
    threadRunning = false;
    std::cout << result << std::endl;
}
```

Dead-code elimination

```cpp
bool threadRunning;

bool proveFermatsLastTheorem() // Thread 1 {
    threadRunning = true;
    while (pow (x, n) + pow (y, n) != pow (z, n)) ++n;

    return false;

}

void testTheorem () {
    bool result;
    startThread ([] () (result = proveFermatsLastTheorem));
    Sleep (2000);
    threadRunning = false;
    std::cout << result << std::endl;
}
```

* if it doesn't call any IO or [[no_return]] is not specified