Top 17 Programming Errors

Acknowledgements to Todd Miller

1. **Not including appropriate header files:** For example, in order to call the sqrt-function the math.h or cmath header file must be included.

2. **Referencing before declaration:** All identifiers must be declared before they are used. Variables for example, must be declared before they can be assigned a value.

3. **Case inconsistency of identifiers:** Remember, the identifier "NewItem" is not the same as "newitem" .

4. **Missing semicolon (or extraneous semicolon):**

   For example, a semi-colon is required after this while-clause:

```
do
   loopBody();
while (condition);
```

   but a semi-colon is NOT necessary after this while-clause (the semi-colon is allowed, but it changes the program logic):

```
while (condition);
   loopBody();
```

5. **Missing the '&' when using stream parameters (i.e., ifstream or ofstream)**

6. **Unexpected timing of type conversion:**

   For example, integer division is performed instead of the desired floating-point division (this is allowed, but precision may be lost because the type conversion did not take place before the division operator was performed):

```
int sum, count;
float average;
   ...
   average = sum / count;
```

7. **Using "=" operator for equality:**

   That's the assignment operator, but you probably want the double equal sign "==" .

8. **Misusing the relational operators within compound conditions:**

   For example, the expression "0 < n < 10" is not logically equivalent to "n > 0 && n < 10" (both expressions are allowed, but the second expression is the proper style).

9. **Badly nested block-statements (brackets):**

   Check them carefully. Better yet, label at least each closing bracket with a comment.

10. **Calling a function without parameters.**

    For example, the following parameterless function call:

```
clrscr;
```

    should have parenthesis around the empty parameter list (this is allowed, but the function will not be called).

11. **Forgetting to declare a parameter as a reference ("&") in a function that is changing the state of that argument**

12. **Failing to initialize variables**

13. **% operator only works with type int**

14. **Your for loop doesn't work correctly?** Check the following error:

```
for (i = 0; i < ct; i + 2)  // this should be i += 2 or i = i + 2
{   }
```

15. Forgetting to put a break in a switch statement

Remember that C does not break out a switch statement if a case is encountered. For example:

```
int x = 2;
switch(x) {
case 2:
  cout << "two\n";
case 3:
  cout << "three\n"
}
```

prints out:

Two
Three

Put a break to break out of the switch:

```
int x = 2;
switch(x) {
case 2:
  cout << "two\n";
  break;
case 3:
  cout << "Three\n";
  break;   /* technically not necessary */
}
```

16. Non-terminated comment, "accidentally" terminated by some subsequent comment, with the code in between swallowed.

```
a=b; /* this is a bug
c=d; /* c=d will never happen */
```

17. Easily changed block scope

```
if( ... )
    foo();
else
    bar();
```

which, when adding debugging statements, becomes

```
if( ... )
    foo(); /* the importance of this semicolon can't be overstated */
else
    cout <<  "Calling bar()\n" ;      /* oops! the else stops here */
    bar();                         /* oops! bar is always executed */
```

There are a large class of similar errors, involving misplaced semicolons and brackets.