1. answer: Any function whose type is not void must use it. A type void function may use it, but no expression is allowed to follow the word “return”.

2. answer: The function cannot modify the actual parameter since it was passed by value and the function is using a copy of it.

3. answer:

```cpp
bool nor(bool a, bool b) {
    if (!a && !b)
        return true;
    else
        return false;
}
```

4. answer:

```cpp
float avg (int a, int b) {
    return ( (a + b)/(float) 2);
}
```

5. answer:

```cpp
int Compare(int a, int b) {
    if (a < b)
        return -1;
    else if (a == b)
        return 0;
    else
        return 1;
}
```

6. answer:

```
10 12
3 12
4 11
4 13
```

7. answer:

```
b = 3
a = 2
a = 0
b = 4
b = 3
```
8. answer:

\[
\begin{align*}
  b &= 3 \\
  a &= 0 \\
  a &= 0 \\
  b &= 4 \\
  b &= 3
\end{align*}
\]

9. answer:

```cpp
#include <iostream>
#include <cmath>
using namespace std;

float calculateQ(float);  // function prototype

int main() {
    float guess, flow, diff;
    cout << "Enter a guess for the depth: ";
    cin >> guess;
    flow = calculateQ(guess);
    diff = (flow - 1000);
    while (fabs(diff) > .001) {
        if (diff < 0)
            cout << "Enter a slightly higher depth: ";
        else
            cout << "Enter a slightly lower depth: ";
        cin >> guess;
        flow = calculateQ(guess);
        diff = (flow - 1000);
    }
    cout << "\nWhen 1000 cubic feet per second of water is " << "\nflowing through the channel, the depth of the " << "\nwater is approximately " << guess << " feet."
    cout << endl;
    return 0;
}

float calculateQ(float d) {
    float hydraulicRadius = (d*15)/(2.0*d+15);
    return (1.49/.014)*15*d*pow(hydraulicRadius,2./3)*sqrt(.0015);
}
```

10. answer: `void f(int &n);`

11. answer:

```
x is 10
y is 20
```

12. answer:

```
5 5
```

13. answer:

```
x is: 4 and y is: 4
z is 9 and x is: 5
```
14. answer:

```
first time
main   funny
x  4   a  &x
y  4   b   7
z  5   c   7

second time
main   funny
x  5   a  &x
y  4   b   9
z  5   c   9
```

15. answer:

By extracting data from a stream or inserting data into a stream, we are changing the stream, and this change must be reflected in the stream object.

16. answer:

Presumably, we want to change the object being passed, namely the SimpleWindow.

17. answer:

```
void ZeroSmaller(int &a, int &b) {
    if (fabs(a) < fabs(b))
        a = 0;
    else
        b=0;
    return;
}
```

18. answer: Its parameter list.

19. answer: The determination of what function to invoke when more than one function has the same name.

20. answer: 2

21. answer: -1

22. answer:

```
b is 5
a is 3
a is 3
b is 5
```
23. answer:
Two semicolons are missing. If these are put in, the program will still not compile because of the redefinition of a default parameter. See the blue box, page 323.

24. answer: 3

25. answer:
void reverse(string s, string &r) {
    int i = s.size()-1;
    r = r + s.substr(i,1);
    s = s.substr(0,s.size()-1);
    if (s.size() > 0)
        reverse(s,r);
}

26. answer:
void PrintNumber(int n) {
    if (n/10 != 0) {
        PrintNumber(n/10);
        cout << n%10;
    } else
        cout << n;
}