Instructions: This homework assignment focuses primarily on some of the basic syntax and semantics of C++. The answers to the following questions can be determined from Chapters 3 and 4 of the lecture notes and Chapters 2 through 4 of the text.

After you have analyzed the questions and decided what answers you believe are correct, you may find it useful to write some short programs to test your logic.

Opscan forms will be passed out in class. Write your name and code your ID number on the opscan form. Turn in your completed opscan at class on Friday June 4.

For questions 1 through 3, select the value of the given C++ arithmetic expression. Note that the presence of a decimal point indicates a `double`, rather than an `int`.

1. \( 6.0 / 8.0 + 5 / 2 \)
   1) 0.0  
   2) 2  
   3) 2.75  
   4) 3.25  
   5) None of these

2. \( 9 / 2 * 2 \)
   1) 2  
   2) 2.25  
   3) 8  
   4) 9  
   5) None of these

3. \( 14 \% 4 + 3 \% 4 \)
   1) 2  
   2) 3  
   3) 4.25  
   4) 5  
   5) None of these

For questions 4 through 6, select the value assigned to the relevant variable, given the declarations:

```cpp
int anInt;
double aDble;
```

4. \( aDble = 4 / 5; \)
   1) 0.0  
   2) 0.8  
   3) 1.0  
   4) This is not allowed  
   5) None of these

5. \( anInt = 4 / 5; \)
   1) 0  
   2) 0.8  
   3) 1  
   4) This is not allowed  
   5) None of these

6. \( aDble = 5 / 2.0; \)
   1) 0.4  
   2) 2.0  
   3) 2.5  
   4) 3.0  
   5) None of these

7. At the hardware level, the values 5 and 5.0 are stored in exactly the same way.
   1) true  
   2) false  
   3) maybe
For questions 8 through 11, assume the following variables have been declared:

```cpp
int anInt;
double aDble;
char aChar;
```

and assume the standard input stream `cin` contains the following values, separated by tabs:

```
1.2 4.5 A -46.32
```

Determine the value of the indicated variable after the execution of the given statement; each question is independent, that is, each starts with the stream contents shown above.

8. `aChar` after `cin >> anInt >> aChar`;
   1) 4  
   2) '4'  
   3) '	' (a tab)  
   4) '.'  
   5) None of these

9. `aChar` after `cin >> aDble >> aChar`;
   1) 4  
   2) '4'  
   3) '	' (a tab)  
   4) '.'  
   5) None of these

10. `anInt` after `cin >> aDble >> anInt`;
    1) 2  
    2) '4'  
    3) 4  
    4) 4.5  
    5) None of these

11. `anInt` after `cin >> anInt; cin.get(aChar); cin >> anInt;`
    1) 1  
    2) 2  
    3) 4  
    4) 5  
    5) None of these

12. What is printed by the statement: `cout << "The answer is" << setw(3) << 30 + 12;`
    1) The answer is 30 + 12  
    2) The answer is 42  
    3) The answer is 42  
    4) The answer is 30 + 12  
    5) None of these

13. Assuming that all variables are of type double, the correct C++ expression for \( \frac{(a+b)c}{d+e} \) is:
    1) \( \frac{a + b * c}{d + e} \)  
    2) \( \frac{(a + b) * c}{d + e} \)  
    3) \( \frac{(a + b) * c}{d + e} \)  
    4) \( \frac{(a + b * c)}{d + e} \)  
    5) None of these
14. What value is assigned to the variable `Average` below?

```cpp
int x = 4, y = 8, z = 5, w = 4;
double Average = (x + y + z + w) / 4;
```

1) 5.25 3) 5.0 5) None of these
2) 5 4) 4.75

15. Given the declaration `int TestScore = 78;`, which of the output statements given below will produce the output:

```
1234567890
Score: 78
```

1) `cout << "1234567890" << endl
   << "Score: " << TestScore << endl;`
2) `cout << "1234567890" << endl
   << "Score:" " " << TestScore << endl;`
3) `cout << "1234567890" << endl
   << "Score:" << setw(4) << TestScore << endl;`

4) All of the above 6) 1 and 3 only 8) None of these
5) 1 and 2 only 7) 2 and 3 only

16. Among the binary C++ operators `+`, `-`, `*`, `/`, and `%`, which have the lowest precedence when an expression is evaluated?

1) `+` and `-`
2) `*` and `/`
3) `,`, `/`, and `%`
4) `+`, `-`, and `%`
5) None of these

17. Suppose that the input stream `cin` contains the IP address: `298.173.41.142`

Assuming that the variables `A` and `B` are declared as `int`s, which of the following code fragments will correctly read the second part of the IP address (173) into the variable `B`?

1) `cin >> A;
cin.ignore(100, '.');
cin >> B;`
2) `cin >> A;
char ch;
cin.get(ch);
cin >> B;`
3) `cin.ignore(100, '.');
cin >> B;`
4) `cin.get(A);
cin.get(B);`
5) All of the above 6) 1 and 2 only 7) 1 and 2 only 8) 1 and 3 only
9) 2 and 3 only 10) None of these

Page 3 of 4
18. A program specification says that a line of input will start with a text label containing up to 25 characters, followed by a tab character, followed by an integer value; for example:

```
Number of nodes:<tab>293
```

Here, <tab> and <newline> indicate the occurrence of a single tab character and a single newline character.

Given the specification, which of the following code fragments will successfully read the integer value into the `int` variable `NetSize`? Assume that `In` is an input file stream variable that has been opened on an input file, and that the data in the stream conforms to the specification.

1) `In.ignore(25, '\t');`  `In >> NetSize;`
2) `In.ignore(30, '\t');`  `In >> NetSize;`
3) `In.ignore(25, ':');`  `In >> NetSize;`
4) `In.ignore(50, '\t');`  `In >> NetSize;`

5) All of the above
6) 1 and 2 only
7) 2 and 4 only
8) 1, 2 and 3 only
9) 1, 2 and 4 only
10) None of these

For questions 19 and 20, assume that the input file `Data.txt` is:

```
1234567890
1234567890
1234567890
1234567890
```

19. What output would the following code fragment produce?

```cpp
#include <iostream>

int main()
{
    std::ifstream In;
    In.open("Data.txt");
    char Value;
    In.ignore(15, '0');
    In >> Value;
    std::cout << "Value: " << Value << std::endl;
    return 0;
}
```

1) Value: 1
2) Value: 2
3) Value: 3
4) Value: 4
5) Value: 5
6) Value: 6
7) Value: 7
8) Value: 8
9) Value: 9
10) Value: 0

20. What output would the following code fragment produce?

```cpp
#include <iostream>

int main()
{
    std::ifstream In;
    In.open("Data.txt");
    char Value;
    In.ignore(5, '\n');
    In.get(Value);
    std::cout << "Value: " << Value << std::endl;
    return 0;
}
```

1) Value: 1
2) Value: 2
3) Value: 3
4) Value: 4
5) Value: 5
6) Value: 6
7) Value: 7
8) Value: 8
9) Value: 9
10) Value: 0