Instructions: This homework assignment focuses on the basics of C++ enums and structs. The answers to the following questions can be determined from Chapters 3 through 10 of the lecture notes and Chapters 10 and 11.1 – 11.2 of the text. Assume any #include directives, variable declarations, etc, which are needed to make the given code syntactically correct.

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.

The on-line Opscan form for this quiz provided by the Curator system must be used for this assignment. (No other submissions will be accepted!) Check the course web site for the due date for this quiz.

The CS1044 online quizzes are open book, open notes, open Web quizzes. It is an honor code violation to discuss, (in any form: written, verbal or electronic), any portion of these quizzes with any other students, (regardless of whether they are taking the course or not). It is also an honor code violation to have a copy of a quiz, (in any form: written, verbal or electronic), in your possession when not taking a quiz. Failure to adhere to any of these restrictions is a Va Tech Honor Code violation, (www.honorsystem.vt.edu).

For questions 1 through 4, consider the declarations:

```cpp
const int MaxEmployees = 1000;
struct Personnel {
    string Name;
    int    IDNum;
    double HourlyPayRate;
};

string Emp5Name = "Joe Bob Hokie";
string Emp7Name = "Haskell Hoo IV";
Personnel Emp5 = {Emp5Name, 401934242, 19.87},
    Emp7 = {Emp5Name, 803291024,  9.32};
Personnel Staff[MaxEmployees];
```

Assume that the array Staff[] has been initialized so that all fields of all the array elements have logically correct values.

1. The name fields of the variables Emp5 and Emp7 could be compared by the expression(s):
   1) Emp5.Name = Emp7.Name
   2) Emp5.Name == Emp7.Name
   3) Emp7[Name] == Emp7[Name]
   4) All of these
   5) 1 and 2 only
   6) 1 and 3 only
   7) 2 and 3 only
   8) None of these

2. The variables Emp5 and Emp7 could be compared by the expression(s):
   1) Emp7 = Emp7
   2) Emp5 == Emp7
   3) Emp5 <= Emp7
   4) All of the above
   5) 1 and 2 only
   6) 1 and 3 only
   7) 2 and 3 only
   8) None of these
3. The statement:  
   ```cpp
   cout << Staff[17].Name;
   ```
   
   1) prints the name of the employee with ID number 17  
   2) prints the name of the employee whose Personnel record is stored at index 17  
   3) is syntactically illegal since you can't dump a struct variable into an output stream  
   4) None of these

4. The statement:  
   ```cpp
   Staff[17] = Emp5;
   ```
   
   1) copies the contents of the variable Emp5 into the Personnel record at index 17  
   2) copies the contents of the Personnel record at index 17 into the variable Emp5  
   3) swaps the contents of the variable Emp5 and the Personnel record at index 17  
   4) is syntactically illegal  
   5) None of these

For questions 5 and 6 assume the following declarations:

```cpp
enum WoodKind {ASH, CEDAR, OAK, PINE, POPLAR, WALNUT};

struct Size {
    int Length;
    int Width;
    int Thickness;
};

struct Board {
    Size      Dimensions;
    WoodKind  Kind;
    int      smoothSurfaces;
};

Board oneBoard = {{2, 4, 8}, PINE, 4};  // Legal initialization.
```

5. The number of smooth surfaces of `oneBoard` could be printed by the statement(s):

   1) `cout << smoothSurfaces;`  
   2) `cout << Wood.smoothSurfaces;`  
   3) `cout << oneBoard.smoothSurfaces;`  
   4) All of these
   5) 1 and 2 only  
   6) 1 and 3 only  
   7) 2 and 3 only  
   8) None of these

6. The width of `oneBoard` could be printed by the statement(s):

   1) `cout << oneBoard.Size.Width;`  
   2) `cout << oneBoard.Dimensions.Width;`  
   3) `cout << oneBoard.Width;`  
   4) All of the above
   5) 1 and 2 only  
   6) 1 and 3 only  
   7) 2 and 3 only  
   8) None of these
For questions 7 and 8 assume the declarations given for questions 14 and 15, and also the declarations:

```cpp
const int MAXBOARDS = 1000;
Board Lumber[MAXBOARDS];
```

Consider implementing a loop to print the dimensions of all OAK boards that occur in the array `Lumber[]`, assuming the array has been initialized to hold data about `numBoards` boards:

```cpp
for (int Idx = 0; Idx < numBoards; Idx++) {
    if (Lumber[Idx].Kind == OAK) {
        cout << "Length: " << Lumber[Idx].Dimensions.Length << endl;
        cout << "Width: " << Lumber[Idx].Dimensions.Width << endl;
        cout << "Thickness: " << Lumber[Idx].Dimensions.Thickness << endl;
    }
}
```

7. How should the first blank in the `if` condition be filled?

1) `Lumber[Idx].Kind` 
2) `Lumber[Idx].WoodKind` 
3) `Kind` 
4) `WoodKind` 
5) None of these

8. How should the second blank in the `if` condition be filled?

1) "OAK" || "oak" 
2) `OAK` 
3) "OAK" 
4) Either 1 or 2 
5) None of these

For questions 9 through 13, assume the type definition:

```cpp
struct Trip {
    string Origin;
    string Destination;
    int Mileage;
    int Time;
    double avgMPH;
};
```

For questions 9 and 10, consider the function definition:

```cpp
// Resets the Time field of a Trip variable, if the given parameter
// is nonnegative.
void setTime(_________ T, int Time) {   // Line 1
    if (Time >= 0) {
        ________ = Time;                // Line 2
    }
}
```

9) How should the blank in Line 1 be filled?

1) `Trip&` 
2) `Trip` 
3) `const Trip&` 
4) Any of these will do 
5) 1 or 2 only 
6) 1 or 3 only 
7) 2 or 3 only 
8) None of these
10) How should the blank in Line 2 be filled?

1) Trip.Time  5) 1 or 2 only
2) T.Time  6) 1 or 3 only
3) Time  7) 2 or 3 only
4) Any of these will do  8) None of these

For questions 11 through 13, consider the function definition:

```cpp
// If the destination of Leg1 is the same as the origin of Leg2, set
// Merged to represent a trip from the origin of Leg1 to the destination
// of Leg2, with the correct total mileage and time.
// Return true if the merge succeeds and false otherwise.

bool mergeTrips(Leg1, Leg2, Merged) {
    if (Leg1.Destination == Leg2.Origin) {
        Merged.Origin = Leg1.Origin;
        Merged.Destination = Leg2.Destination.
        Merged.Mileage = Leg1.Mileage + Leg2.Mileage;
        Merged.Time = Leg1.Time + Leg2.Time;
        return true;
    }
    return false;
}
```

11) How should the **first** blank in the formal parameter list be filled?

1) Trip&  5) 1 or 2 only
2) Trip  6) 1 or 3 only
3) const Trip&  7) 2 or 3 only
4) Any of these will do  8) None of these

12) How should the **third** blank in the formal parameter list be filled?

1) Trip&  5) 1 or 2 only
2) Trip  6) 1 or 3 only
3) const Trip&  7) 2 or 3 only
4) Any of these will do  8) None of these

13) How should the blank in the if-condition be filled?

1) Leg1.Destination == Leg2.Origin
2) Origin == Destination
3) Leg1 == Leg2
4) Leg1.Origin == Leg2.Destination
5) None of these
On-line submission of a quiz constitutes your Virginia Tech Honor Code Pledge:

Virginia Tech Honor Code:

"On my honor, I have neither given nor received unauthorized aid on this examination."