

Given this partial class:

```
class Course
{
    public:
        int getCreditHours();
        float getLabFee();
        // other function prototypes will go here
    private:
        string dept;           // department offering the course, like CSci
        int number;           // course number, like 270
        int creditHours;      // # of credit hours a student earns, like 3
        bool semOffered[8];   // semesters this course is usually offered where
                               // subscript 1 = spring, 3 = summerI, 4 = summerII, 7 = fall
        float labFee;        // amount charged for a lab fee, like 25.00
}
```

1. What would be the name of a constructor function for this class? _____

A constructor function always has the same name as the class:

```
Course(); // default constructor for this class
Course(int number, int creditHours, float labFee);
```

Also constructors have no type. A constructor is neither a value-returning function nor a void function.

2. A client program using this class could legally use this declaration:

```
int dept;
```

yes no

Yes, a client program may declare variables which have the same names as private variables in a class included by the program. Since the private variables are not directly accessible by the client program, there is no conflict. In fact, even if dept was a public variable of the Course class there would still be no conflict as class variables are part of and accessed through instances of the Course objects.

3. Show how a client would declare an array of 10 Course variables.

```
Course courses[10];
```

4. Write the implementation for a class function (as it would appear in the class implementation file) which will take an integer parameter 1-8 and return true if the course is usually offered in that semester or false if it is not, or if the parameter is not a valid semester.

```
bool Course::isOffered(int semester)
{
    return semOffered[semester];
}
```

or

```
bool Course::isOffered(int semester)
{
    if (semOffered[subscript] == true)
        return true;
    else
        return false;
}
```

5. Given these declarations: Course c;
 int s;

Show how a client would call the function you wrote for #4 to find out if Course c is being offered in semester s (assume appropriate values have been assigned to c and s). Print YES if the course is usually offered or NO if it is not.

```
if (c.isOffered(s))
    cout << "YES" << endl;
else
    cout << "NO" << endl;
```