

NET2006: Sample Quiz

Assume you have the following **class** specification:

```
class Telephone {
private:
    int nAreaCode;    // 3-digit area code
    int nLocal;      // 7-digit phone number
public:
    // more code in this section of the class specification
};
```

- The code shown above:
 - Causes exactly 8 bytes of space to be allocated for an object called **Telephone**.
 - Generates allocation in *stack* memory.
 - By itself, does not cause any space to be allocated.
 - Has one or more syntax errors.
 - none of the above.
- A suitable constructor would be:
 - `void Telephone() { nAreaCode = 613; nLocal = 0; }`
 - `Telephone() : nAreaCode(613), nLocal(0) { }`
 - `bool Telephone() { nAreaCode = 613; nLocal = 0; return true; }`
 - `Telephone() = 0;`
 - none of the above.
- Create a **Telephone** object called **Home**.
 - `Home Telephone;`
 - `Telephone Home;`
 - `Telephone Home() = 0;`
 - `Telephone Home(613, 7274723);`
 - none of the above.
- Identify the member function that can set the values in a **Telephone** object:
 - `void Set(int nAreaCode, int nLocal) { }`
 - `void Set(int nAC, int nLN) { nAreaCode = nAC; nLocal = nLN; }`
 - `void Set(int nAC, int nLN) : nAreaCode(nAC), nLocal(nLN) { }`
 - `void Set(int nAreaCode, int nLocal) { nAreaCode = nAreaCode; nLocal = nLocal; }`
 - none of the above.
- How many bytes will **Home** consume (in our WindowsXP operating environment).
 - 2
 - 4
 - 10
 - 8
 - 12
- Set the telephone number for **Home** to (613) 555-1212.
 - `Home.Set(613, 555-1212);`
 - `Set.Home(613, 555-1212);`
 - `Home = { 613, 5551212 };`
 - `Home.Set("613 ", "555-1212 ");`
 - none of the above.
- The dot operator:
 - Is used to point from one member to the next.
 - Associates one **struct** with another **struct**.
 - Is called the scope resolution operator.
 - Is called the binding operator.
 - none of the above.
- In which way does a **class** differ from a **struct**.
 - By default all members of a **class** are **private**; all members of a **struct** are **public**.
 - By default all members of a **class** are **public**; all members of a **struct** are **private**.
 - A **class** can have data and functions as members; a **struct** can have only data as members.
 - Objects created from a **class** are allocated on the **stack**; objects created from a **struct** are allocated on the **heap**.
 - none of the above.
- The keyword **this**:
 - Constructs an object with default initialization values.
 - Is used to call member functions.
 - Identifies the name of the class at run-time.
 - Holds the address of the object that called a member function.
 - none of the above.
- Assume you have a bigger **for** loop (though I only show the first part of the **for** statement). Which of the following correctly counts from 1 to 10:
 - `for (int i=1; i<11; i=i+1)`
 - `for (int i=0; i<10; i=i+1)`
 - `for (int i=0; i<10; i++)`
 - `for (int i=1; i<=10; i++)`
 - `for (int i=0, i<=10, i=i+1)`