

1. Circle the following statements as true or false.

- T  F The data members of a class must be of the same type.
- T  F The function members of a class must be *public*.
- T F A class can have more than one constructor.
- T F The only built-in operations on classes are the assignment (=) and member access (.) operations.
- T F A user-defined class can be used as the return type for a function.
- T  F A user-defined class can not be passed by reference as a parameter to a function.
- T  F Constructors are declared using the *constructor* keyword.
- T F Constructors and destructors are functions without any return type.
- T F Private members cannot be accessed by a user of the class.
- T F A member of a class can be a function or a variable (that is data).

2. Given this C++ class specification:

```
class TelNum
{
public:
    void PrintTelNum ();           // prints to the screen like: 903-886-5409
    void SetTelNum (int ac, int exch, int ext); // assigns parameter values
    bool SameNum (TelNum otherNum); // returns true if two numbers are the same,
    int GetAreaCode ();           // returns area code
    int GetExchange ();           // returns exchange
    int GetExtension ();          // returns extension
private:                          // for 903-886-5409
    int areaCode;                 // area code is 903
    int exchange;                 // exchange is 886
    int extension;                // extension is 5409
};
```

- a) Show how the client could declare class variables called `myNum` and `yourNum` to hold telephone numbers using the `TelNum` class.

`TelNum myNum, yourNum;`

- b) Show how the client could assign the number 903-886-5409 to `myNum`.

`myNum.SetTelNum(903, 886, 5409);`

- c) Show how the client could call the SameNum function and print YES if myNum is the same as yourNum or print NO otherwise (assume both variables are assigned values).

```
if (myNum.SameNum(yourNum))
    cout << "YES" << endl;
else
    cout << "NO" << endl;
```

- e) Show how the client could check to see if the area code of myNum is 903 (print YES if it is or NO if it's not).

```
if (myNum.GetAreaCode() == 903)
    cout << "YES" << endl;
else
    cout << "NO" << endl;
```

- f) Write the implementation (body) for the class function SetTelNum as it would appear in the implementation file.

```
void TelNum::SetTelNum(int ac, int exch, int ext)
{
    areaCode = ac;
    exchange = exch;
    extension = ext;
}
```

- g) Assuming that the client has no other variables declared other than myNum and yourNum, which of the following statements could the client code in a client program?

- yes  no 1) cout << myNum.areaCode;
- yes no 2) yourNum = myNum;
- yes  no 3) if (myNum == yourNum) cout << "same number";
- yes no 4) int areaCode;
- yes  no 5) cout << myNum;
- yes no 6) void DoSomething (TelNum & t); // client-defined function
- yes no 7) TelNum DoSomethingElse (TelNum t); // client-defined function
- yes  no 8) extension = 5494;

h) What difference (if any) does it make to the client if the keyword "private:" is removed from the class declaration (and everything else remains the same)?

The member variables (areaCode, exchange and extension) become public. This will mean that they can be directly changed by the client as they are no longer private. Will still work but might be dangerous, breaks encapsulation of the implementation of the TelNum class.

i) What difference (if any) does it make to the client if the keyword "public:" is removed from the class declaration (and everything else remains the same)?

Things are private by default in user defined classes. So without the "public" declaration, all of the member functions become private. This makes the class pretty useless as the client can no longer do anything with the class. All member functions and member variables are private so the client can no longer call them or access any of them.