**Chapter 16: Introduction to Classes, Objects and Strings**

**Section 16.2 Defining a Class with a Member Function**

16.2 Q1: C++ functions other than main are executed:

1. Before main executes.
2. After main completes execution.
3. When they are explicitly called by another function.
4. Never.

**ANS c. When they are explicitly called by another function.**

16.2 Q2: Calling a member function of an object requires which item?

1. The dot operator.
2. Open and close braces.
3. The class name.
4. None of the above.

**ANS a. The dot operator.**

16.2 Q3: In the UML, the top compartment of the rectangle modeling a class contains:

1. The class’s name.
2. The class’s attributes.
3. The class’s behaviors.
4. All of the above.

**ANS: a. The class’s name.**

**Section 16.3 Defining a Member Function with a Parameter**

16.3 Q1: What is the name of the values the method call passes to the method for the parameters?

1. Arguments.
2. References.
3. Objects.
4. Values.

**ANS: a. Arguments.**

16.3 Q2: Assuming that text is a variable of type string, what will be the contents of text after the statement cin >> text; is executed if the user types Hello World! then presses *Enter*?

1. "H"
2. "Hello"
3. "Hello World"
4. "Hello World!"

**ANS: b. Hello**

16.3 Q3: Two adjacent parameters are separated by what symbol?

1. Dot.
2. Comma.
3. Parentheses.
4. Braces.

**ANS: b. Comma.**

**Section 16.4 Data Members, *set* Member Functions and *get* Member Functions**

16.4 Q1: Attributes of a class are also known as:

1. Constructors.
2. Local variables.
3. Data members.
4. Classes.

**ANS: c. Data members.**

16.4 Q2: What is the default initial value of a String?

1. ""
2. "default"
3. default
4. None of the above.

**ANS: a. ""**

16.4 Q3: What type of member functions allow a client of a class to assign values to private data members?

1. *Client* member functions.
2. *Get* member functions.
3. *Set* member functions.
4. None of the above.

**ANS: c. *Set* member functions.**

**Section 16.5 Initializing Objects with Constructors**

16.5 Q1: A *default* constructor has how many parameters?

1. 0.
2. 1.
3. 2.
4. Variable number.

**ANS: a. 0.**

16.5 Q2: A constructor can specify the return type:

1. int.
2. string.
3. void.
4. A constructor cannot specify a return type.

**ANS: d. A constructor cannot specify a return type.**

16.5 Q3: The compiler will *implicitly* create a default constructor if:

1. The class does not contain any data members.
2. The programmer specifically requests that the compiler do so.
3. The class does not define any constructors.
4. The class already defines a default constructor.

**ANS: c. The class does not define any constructors.**

16.5 Q4 [C++11]: Which of the following is *true*?

1. The only way to define a constructor in a class is to explicitly define one.
2. If you define any constructors with arguments, the compiler will also define a default constructor.
3. If you define any constructors with arguments, the compiler will not define a default constructor.
4. You cannot explicitly create constructors.

**ANS: c. If you define any constructors with arguments, the compiler will not define a default constructor.**

**Section 16.6 Placing a Class in a Separate File for Reusability**

16.6 Q1: A header file is typically given the filename extension:

1. .h
2. .hdr
3. .header
4. .cpp

**ANS: a. .h**

16.6 Q2: Assuming that GradeBook.h is found in the current directory and the iostream header file is found in the C++ Standard Library header file directory, which of the following preprocessor directives will *fail* to find its desired header file?

1. #include <iostream>
2. #include "iostream"
3. #include <GradeBook.h>
4. #include "GradeBook.h"

**ANS: c. #include <GradeBook.h>**

**Section 16.7 Separating Interface from Implementation**

16.7 Q1: In the source-code file containing a class’s member function definitions, each member function definition must be tied to the class definition by preceding the member function name with the class name and ::, which is known as the:

1. Member definition linker.
2. Class implementation connector.
3. Source code resolver.
4. Scope resolution operator.

**ANS: d. Scope resolution operator.**

16.7 Q2: When compiling a class’s source code file (which does not contain a main function), the information in the class’s header file is used for *all* of the following, *except*:

1. Ensuring that the header of each member function matches its prototype.
2. Ensuring that each member function knows about the class’s data members and other member functions.
3. Determining the correct amount of memory to allocate for each object of the class.
4. All of the above are uses that the compiler has for the header file information.

**ANS: d. All of the above are uses that the compiler has for the header file information.**

16.7 Q3: When a client code programmer uses a class whose implementation is in a separate file from its interface, that implementation code is merged with the client’s code during the:

1. Programming phase.
2. Compiling phase.
3. Linking phase.
4. Executing phase.

**ANS: c. Linking phase.**

**Section 16.8 Validating Data with *set* Functions**

16.8 Q1: To execute multiple statements when an if statement’s condition is true, enclose those statements in a pair of:

1. Parentheses, ( ).
2. Square Brackets, [ ].
3. Braces, { }.
4. Angle brackets, < >.

**ANS: c. Braces, { }.**

16.8 Q2: Assuming that the string object text contains the string "Hello!!! ", the "expression text.substr( 2 , 5 ) would return a string object containing the string:

1. "llo!! ".
2. "llo! ".
3. "ello! ".
4. "ello".

**ANS: a. "llo!!** **".**