|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Machine language is expressed as a series of 1s and 0s.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2. C# programmers must use Pascal casing when creating method names to produce an executable program.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | False | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3. The C# programming language was developed as an object-oriented and component-oriented language.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4. When the keyword void is used in the Main() method header, it indicates that the Main() method is empty.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | False | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5. The Visual Studio IDE gives you advanced features such as syntax coloring and automatic statement completion.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6. Internally, computers are constructed from circuitry that consists of small on/off switches. What is the most basic circuitry-level language that computers use to control the operation of those switches called?   |  |  |  | | --- | --- | --- | |  | a. | syntax | |  | b. | machine language | |  | c. | compiler | |  | d. | program |  |  |  | | --- | --- | | *ANSWER:* | b | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7. What kind of programming language allows you to use a vocabulary of reasonable terms such as "read," "write," or "add" instead of the sequence of on/off switches that perform these tasks?   |  |  |  | | --- | --- | --- | |  | a. | high-level | |  | b. | machine-level | |  | c. | low-level | |  | d. | switch-level |  |  |  | | --- | --- | | *ANSWER:* | a | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8. Programmers make use of what type of program in order to translate higher-level language statements into machine code?   |  |  |  | | --- | --- | --- | |  | a. | A command prompt | |  | b. | An IDE | |  | c. | A compiler | |  | d. | A JIT |  |  |  | | --- | --- | | *ANSWER:* | c | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9. A program's execution of various statements and procedures in a correct order to produce desired results is referred to as what defining characteristic?   |  |  |  | | --- | --- | --- | |  | a. | The program's GUI. | |  | b. | The attributes utilized by the program. | |  | c. | The methods of the program. | |  | d. | The program's logic. |  |  |  | | --- | --- | | *ANSWER:* | d | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10. What is the process of removing all syntax and logical errors from a program in order to create a working program that accomplishes all intended tasks known as?   |  |  |  | | --- | --- | --- | |  | a. | debugging | |  | b. | compiling | |  | c. | commenting out | |  | d. | executing |  |  |  | | --- | --- | | *ANSWER:* | a | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11. What type of program is created by the use of named memory locations and a series of steps or operations to manipulate the values of those memory locations?   |  |  |  | | --- | --- | --- | |  | a. | object-oriented | |  | b. | component-oriented | |  | c. | event-driven | |  | d. | procedural |  |  |  | | --- | --- | | *ANSWER:* | d | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 12. What term describes a one-word name with no embedded spaces that references a variable in a program?   |  |  |  | | --- | --- | --- | |  | a. | behavior | |  | b. | bug | |  | c. | identifier | |  | d. | attribute |  |  |  | | --- | --- | | *ANSWER:* | c | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13. A series of four or five comparisons and calculations that together determine an employee's withholding tax value might be grouped using what sort of logical unit?   |  |  |  | | --- | --- | --- | |  | a. | an attribute | |  | b. | a method | |  | c. | a class | |  | d. | a structure |  |  |  | | --- | --- | | *ANSWER:* | b | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 14. What programming style capitalizes the first letter of all new words in an identifier, including the first one?   |  |  |  | | --- | --- | --- | |  | a. | camel casing | |  | b. | upper casing | |  | c. | OOP casing | |  | d. | Pascal casing |  |  |  | | --- | --- | | *ANSWER:* | d | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15. What is an object in relation to a defined class in a programming language?   |  |  |  | | --- | --- | --- | |  | a. | It is a property of the class. | |  | b. | It is an interface to the class. | |  | c. | It is an instance of the class. | |  | d. | It defines the class state. |  |  |  | | --- | --- | | *ANSWER:* | c | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16. What technique involves the packaging of an object's attributes and methods into a cohesive unit that can be used as an undivided entity?   |  |  |  | | --- | --- | --- | |  | a. | Encapsulation | |  | b. | Polymorphism | |  | c. | Inheritance | |  | d. | Interface |  |  |  | | --- | --- | | *ANSWER:* | a | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 17. Encapsulation is similar to using a device you use without regard for the internal mechanisms. What is the common term for this type of device?   |  |  |  | | --- | --- | --- | |  | a. | white box | |  | b. | black box | |  | c. | gray box | |  | d. | blue box |  |  |  | | --- | --- | | *ANSWER:* | b | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 18. What can be used to extend an existing class so as to create a more specific class?   |  |  |  | | --- | --- | --- | |  | a. | Inheritance | |  | b. | Encapsulation | |  | c. | Polymorphism | |  | d. | Abstraction |  |  |  | | --- | --- | | *ANSWER:* | a | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 19. What information must be supplied when utilizing a method that requires additional information in order to operate?   |  |  |  | | --- | --- | --- | |  | a. | Literal strings | |  | b. | Primitive data | |  | c. | Arguments | |  | d. | Complex data |  |  |  | | --- | --- | | *ANSWER:* | c | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20. The program you are creating must output information onto the screen, and then position the cursor on the next line in preparation for additional output. What method should you use?   |  |  |  | | --- | --- | --- | |  | a. | Write() | |  | b. | Println() | |  | c. | PrintLine() | |  | d. | WriteLine() |  |  |  | | --- | --- | | *ANSWER:* | d | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21. What can be used as a construct that acts like a container to provide a way to group similar classes?   |  |  |  | | --- | --- | --- | |  | a. | namespace | |  | b. | method | |  | c. | object | |  | d. | black box |  |  |  | | --- | --- | | *ANSWER:* | a | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 22. The use of "void" and "static" are both examples of what predefined C# language component?   |  |  |  | | --- | --- | --- | |  | a. | classes | |  | b. | attributes | |  | c. | objects | |  | d. | keywords |  |  |  | | --- | --- | | *ANSWER:* | d | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 23. What are nonexecuting statements that you can use to document or add notes to assist in the use of the program?   |  |  |  | | --- | --- | --- | |  | a. | Aliases | |  | b. | Program comments | |  | c. | Namespaces | |  | d. | Whitespaces |  |  |  | | --- | --- | | *ANSWER:* | b | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 24. What must be done once a C# program has been finished before it can be used?   |  |  |  | | --- | --- | --- | |  | a. | It must be fed to a runtime interpreter. | |  | b. | It must be compiled into intermediate language. | |  | c. | It must be stripped of all comments before it can run. | |  | d. | It must be passed through a garbage collector to reduce unnecessary instructions. |  |  |  | | --- | --- | | *ANSWER:* | b | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 25. Where can a method's name and information about what will be passed into and returned from a method be found?   |  |  |  | | --- | --- | --- | |  | a. | In the body of the method. | |  | b. | In the method's keywords. | |  | c. | In the method's class definitions. | |  | d. | In the method's header. |  |  |  | | --- | --- | | *ANSWER:* | d | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 26. An identifier that is prefixed with an @ and allows you to use code written in other languages that do not have the same set of reserved keywords is known by what name?   |  |  |  | | --- | --- | --- | |  | a. | A language identifier | |  | b. | A verbatim identifier | |  | c. | A translating identifier | |  | d. | A commenting identifier |  |  |  | | --- | --- | | *ANSWER:* | b | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 27. What command can be used from the Developer Command Prompt in order to compile a C# program?   |  |  |  | | --- | --- | --- | |  | a. | c#-compile | |  | b. | csc | |  | c. | gcc-c# | |  | d. | csharpc |  |  |  | | --- | --- | | *ANSWER:* | b | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 28. When writing C# code, how do you indicate a namespace?   |  |  |  | | --- | --- | --- | |  | a. | You must specify the namespace by using the "namespace" keyword, followed by the namespace. | |  | b. | You must use the "declare namespace" keywords, followed by the namespace. | |  | c. | You must utilize the using clause, or using directive, by specifying "using", followed by the namespace. | |  | d. | You must use create a file called "namespaces", and include all namespace code in this file. Then, the namespace can be used by name in your program. |  |  |  | | --- | --- | | *ANSWER:* | c | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 29. What is NOT considered a keyword by the C# language?   |  |  |  | | --- | --- | --- | |  | a. | implicit | |  | b. | catch | |  | c. | static | |  | d. | global |  |  |  | | --- | --- | | *ANSWER:* | d | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 30. What statement regarding the C# programming language is accurate?   |  |  |  | | --- | --- | --- | |  | a. | The C# programming language was developed as a procedural language. | |  | b. | C# only allows specific pieces of data to be treated as an object. | |  | c. | C# provides constructs for creating components with properties, methods, and events. | |  | d. | C# is modeled after the COBOL programming language. |  |  |  | | --- | --- | | *ANSWER:* | c | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Match each item with a statement below. Match each item with a statement below:   |  |  | | --- | --- | | a. | syntax | | b. | logical error | | c. | command line | | d. | command prompt | | e. | state of an object | | f. | interface | | g. | literal string | | h. | inheritance | | i. | whitespace | | j. | just in time compiler | |

|  |  |  |
| --- | --- | --- |
| 31. The rules of a high-level programming language   |  |  | | --- | --- | | *ANSWER:* | a | |

|  |  |  |
| --- | --- | --- |
| 32. Adding when you should be multiplying   |  |  | | --- | --- | | *ANSWER:* | b | |

|  |  |  |
| --- | --- | --- |
| 33. The line on which you type a command in a system that uses a text interface   |  |  | | --- | --- | | *ANSWER:* | c | |

|  |  |  |
| --- | --- | --- |
| 34. A request for input that appears at the beginning of the command line   |  |  | | --- | --- | | *ANSWER:* | d | |

|  |  |  |
| --- | --- | --- |
| 35. The value of an object's attributes at any point in time   |  |  | | --- | --- | | *ANSWER:* | e | |

|  |  |  |
| --- | --- | --- |
| 36. The description of interaction between a method and an object   |  |  | | --- | --- | | *ANSWER:* | f | |

|  |  |  |
| --- | --- | --- |
| 37. A series of characters that will be used exactly as entered   |  |  | | --- | --- | | *ANSWER:* | g | |

|  |  |  |
| --- | --- | --- |
| 38. The ability to extend a class to create a more specific class   |  |  | | --- | --- | | *ANSWER:* | h | |

|  |  |  |
| --- | --- | --- |
| 39. Any combination of spaces, tabs, and carriage returns (blank lines)   |  |  | | --- | --- | | *ANSWER:* | i | |

|  |  |  |
| --- | --- | --- |
| 40. Translates intermediate code into executable code   |  |  | | --- | --- | | *ANSWER:* | j | |

|  |  |  |
| --- | --- | --- |
| 41. What are the attributes and state of an object?   |  |  | | --- | --- | | *ANSWER:* | Object-oriented programming (OOP) is an extension of procedural programming. OOP uses variables and methods like procedural programs do, but it focuses on objects. An object is a concrete entity that has attributes and behaviors. The attributes of an object are the features it "has"; the values of an object's attributes constitute the state of the object. For example, attributes of a paycheck include its payee and monetary value, and the state of those attributes might be "Alice Nelson" and $400. | |

|  |  |  |
| --- | --- | --- |
| 42. Explain the concept of methods in object-oriented programming.   |  |  | | --- | --- | | *ANSWER:* | For convenience, the individual operations used in a computer program often are grouped into logical units called methods. For example, a series of four or five comparisons and calculations that together determine an employee's federal tax withholding value might be grouped as a method named CalculateFederalWithholding(). | |

|  |  |  |
| --- | --- | --- |
| 43. What are the features supported by object-oriented programming?   |  |  | | --- | --- | | *ANSWER:* | For a language to be considered object-oriented, it must support the following features: \* Classes \* Objects \* Encapsulation and Interfaces \* Inheritance \* Polymorphism | |

|  |  |  |
| --- | --- | --- |
| 44. Explain the main characteristics of encapsulation.   |  |  | | --- | --- | | *ANSWER:* | Like procedural programs, object-oriented programs have variables (attributes) and procedures (methods), but the attributes and methods are encapsulated into objects that are then used much like real-world objects. Encapsulation is the technique of packaging an object's attributes and methods into a cohesive unit that can be used as an undivided entity. Programmers sometimes refer to encapsulation as using a "black box," a device you use without regard for the internal mechanisms. If an object's methods are well written, the user is unaware of the low-level details of how the methods are executed; in such a case, the user must understand only the interface or interaction between the method and object. For example, if you can fill your Automobile with gasoline, it is because you understand the interface between the gas pump nozzle and the vehicle's gas tank opening. You don't need to understand how the pump works or where the gas tank is located inside your vehicle. If you can read your speedometer, it does not matter how the display figure is calculated. In fact, if someone produces a new, more accurate speedometer and inserts it into your Automobile, you don't have to know or care how it operates, as long as the interface remains the same as the previous one. The same principles apply to well-constructed objects used in object-oriented programs. | |

|  |  |  |
| --- | --- | --- |
| 45. Explain the main characteristics of inheritance.   |  |  | | --- | --- | | *ANSWER:* | Inheritance provides the ability to extend a class so as to create a more specific class. The more specific class contains all the attributes and methods of the more general class, and usually contains new attributes or methods as well. For example, if you have created a Dog class, you might then create a more specific class named ShowDog. Each instance of the ShowDog class would contain all the attributes and methods of a Dog, along with additional methods or attributes. | |

|  |  |  |
| --- | --- | --- |
| 46. How would you compare C# with Java?   |  |  | | --- | --- | | *ANSWER:* | C# is very similar to Java because Java was also based on C++. However, C# is more truly object-oriented. Unlike in Java, every piece of data in C# is an object, providing all data with increased functionality. In Java, simple data types are not objects; therefore, they do not work with built-in methods. Additionally, in Java, data can only be passed to and from methods using a copy; C# omits this limitation. | |

|  |  |  |
| --- | --- | --- |
| 47. What are the components of a C# method?   |  |  | | --- | --- | | *ANSWER:* | Every method in C# contains a header and a body. A method header includes the method name and information about what will pass into and be returned from a method. The method body of every method is contained within a pair of curly braces and includes all the instructions executed by the method. | |

|  |  |  |
| --- | --- | --- |
| 48. What is the meaning of the keyword static in C# in a method header?   |  |  | | --- | --- | | *ANSWER:* | In C#, the reserved keyword static indicates that the method will be executed through a class-not by a variety of objects. It means that you do not need to create an object to use the method; rather, the method is invoked using the class name. | |

|  |  |  |
| --- | --- | --- |
| 49. What are the requirements when choosing an identifier for a C# class?   |  |  | | --- | --- | | *ANSWER:* | You can define a C# class using any identifier you need, as long as it meets the following requirements: \* An identifier must begin with an underscore, the "at" sign (@), or a letter. (Letters include foreign-alphabet letters, which are contained in the set of characters known as Unicode.) \* An identifier can contain only letters, digits, underscores, and the "at" sign (@). It cannot contain spaces or any other punctuation or special characters such as #, $, or &. \* An identifier cannot be a C# reserved keyword, such as class or void. (Actually, you can use a keyword as an identifier if you precede it with an "at" sign, as in @class. An identifier with an @ prefix is a verbatim identifier. This feature allows you to use code written in other languages that do not have the same set of reserved keywords. However, when you write original C# programs, you should not use the keywords as identifiers.) | |

|  |  |  |
| --- | --- | --- |
| 50. What are the types of comments supported by C#?   |  |  | | --- | --- | | *ANSWER:* | There are three types of comments in C#: \* Line comments start with two forward slashes (//) and continue to the end of the current line. Line comments can appear on a line by themselves, or they can occupy part of a line following executable code. \* Block comments start with a forward slash and an asterisk (/\*) and end with an asterisk and a forward slash (\*/ ). Block comments can appear on a line by themselves, on a line before executable code, or after executable code. When a comment is long, block comments can extend across as many lines as needed. \* C# also supports a special type of comment used to create documentation from within a program. These comments, called XML-documentation format comments, use a special set of tags within angle brackets (<>). (XML stands for Extensible Markup Language.) | |