|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Software can be classified into two broad types: application software and programming software.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | False | | *FEEDBACK:* | |  |  | | --- | --- | | *Correct* | Correct. Software can be classified into application software and system software. | | *Incorrect* | Incorrect. Software can be classified into application software and system software. | | | *POINTS:* | 1 | | *REFERENCES:* | 2 Understanding Computer Systems | | *QUESTION TYPE:* | True / False | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.01 - Describe computer systems | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2. Every programming language has rules governing its word usage and punctuation.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | | *FEEDBACK:* | |  |  | | --- | --- | | *Correct* | Correct. Programming languages use a compiler or interpreter to convert source code into machine language. The compiler or interpreter needs to have correct syntax to correctly translate the program into machine language. Correct syntax follows a specific set of rules defining word usage and punctuation. | | *Incorrect* | Incorrect. Programming languages use a compiler or interpreter to convert source code into machine language. The compiler or interpreter needs to have correct syntax to correctly translate the program into machine language. Correct syntax follows a specific set of rules defining word usage and punctuation. | | | *POINTS:* | 1 | | *REFERENCES:* | 3 Understanding Computer Systems | | *QUESTION TYPE:* | True / False | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.01 - Describe computer systems | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3. Besides the popular, comprehensive programming languages such as Java and C++, many programmers use scripting languages such as Python, Lua, Perl, and PHP.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | | *FEEDBACK:* | |  |  | | --- | --- | | *Correct* | Correct. Scripting languages such as Python, Lua, Perl, and PHP can be typed directly from a keyboard and are stored as text rather than as binary executable files, from which they are interpreted line by line when the program executes, rather than compiled to machine language (binary). | | *Incorrect* | Incorrect. Scripting languages such as Python, Lua, Perl, and PHP can be typed directly from a keyboard and are stored as text rather than as binary executable files, from which they are interpreted line by line when the program executes, rather than compiled to machine language (binary). | | | *POINTS:* | 1 | | *REFERENCES:* | 4 Understanding Computer Systems | | *QUESTION TYPE:* | True / False | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.01 - Describe computer systems | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4. Professional computer programmers write programs to satisfy their own needs.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | False | | *FEEDBACK:* | |  |  | | --- | --- | | *Correct* | Correct. Professional programmers write programs to satisfy the needs of others, called users or end users. | | *Incorrect* | Incorrect. Professional programmers write programs to satisfy the needs of others, called users or end users. | | | *POINTS:* | 1 | | *REFERENCES:* | 8 Understanding the Program Development Cycle | | *QUESTION TYPE:* | True / False | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5. The heart of the programming process lies in planning the program’s logic.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | | *FEEDBACK:* | |  |  | | --- | --- | | *Correct* | Correct. The first step of programming is to understand the needs of the end users, but the heart of programming lies in planning the program’s logic. It is during this phase that the programmer plans the steps of the program and how to order them. | | *Incorrect* | Incorrect. The first step of programming is to understand the needs of the end users, but the heart of programming lies in planning the program’s logic. It is during this phase that the programmer plans the steps of the program and how to order them. | | | *POINTS:* | 1 | | *REFERENCES:* | 9 Planning the Logic | | *QUESTION TYPE:* | True / False | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6. Once a program is completely coded, it is ready for a company or organization to use.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | False | | *FEEDBACK:* | |  |  | | --- | --- | | *Correct* | Correct. Coding is only one stage of the product development cycle. Once a program is coded and the syntax is checked for errors, it needs to be tested for logical errors. The process of finding and correcting program errors is called debugging, and programs should only be put into production after they are debugged and tested. | | *Incorrect* | Incorrect. Coding is only one stage of the product development cycle. Once a program is coded and the syntax is checked for errors, it needs to be tested for logical errors. The process of finding and correcting program errors is called debugging, and programs should only be put into production after they are debugged and tested. | | | *POINTS:* | 1 | | *REFERENCES:* | 13 Putting the Program into Production | | *QUESTION TYPE:* | True / False | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7. Pseudocode is an English-like representation of the logical steps necessary to solve a problem.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | | *FEEDBACK:* | |  |  | | --- | --- | | *Correct* | Correct. An English-like representation of the logical steps it takes to solve a problem is called pseudocode. "Pseudo" is a prefix that means "false," so "pseudocode" simply means "false code," or sentences that appear like programming language but do not follow syntax rules. | | *Incorrect* | Incorrect. An English-like representation of the logical steps it takes to solve a problem is called pseudocode. "Pseudo" is a prefix that means "false," so "pseudocode" simply means "false code," or sentences that appear like programming language but do not follow syntax rules. | | | *POINTS:* | 1 | | *REFERENCES:* | 15 Using Pseudocode Statements and Flowchart Symbols | | *QUESTION TYPE:* | True / False | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.04 - Write pseudocode statements and draw flowchart symbols | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8. An infinite loop is a flow of program logic that repeats and never ends.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | | *FEEDBACK:* | |  |  | | --- | --- | | *Correct* | Correct. The repetition of a series of steps is called a *loop*. If the logic in a loop contains a problem such that it never exits the loop, the situation is known as an *infinite loop*. | | *Incorrect* | Incorrect. The repetition of a series of steps is called a *loop*. If the logic in a loop contains a problem such that it never exits the loop, the situation is known as an *infinite loop*. | | | *POINTS:* | 1 | | *REFERENCES:* | 19 Repeating Instructions | | *QUESTION TYPE:* | True / False | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.04 - Write pseudocode statements and draw flowchart symbols | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9. Alan Turing is often regarded as the first programmer.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | False | | *FEEDBACK:* | |  |  | | --- | --- | | *Correct* | Correct. Ada Lovelace predicted the development of software in 1843; she is often regarded as the first programmer. Alan Turing proposed the basis for modern software in 1935 when he theorized a universal Turing machine that could be programmed to follow a set of instructions. | | *Incorrect* | Incorrect. Ada Lovelace predicted the development of software in 1843; she is often regarded as the first programmer. Alan Turing proposed the basis for modern software in 1935 when he theorized a universal Turing machine that could be programmed to follow a set of instructions. | | | *POINTS:* | 1 | | *REFERENCES:* | 27 Understanding the Evolution of Programming Models | | *QUESTION TYPE:* | True / False | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.07 - Describe the evolution of programming models | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10. Programmers can use either procedural programming or object-oriented programming to develop programs.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | | *FEEDBACK:* | |  |  | | --- | --- | | *Correct* | Correct. Modern programmers use two major models or paradigms to develop programs and their procedures. Procedural programming focuses on procedures or actions that are carried out in the program. Object-oriented programming focuses on objects and their attributes and behaviors. | | *Incorrect* | Incorrect. Modern programmers use two major models or paradigms to develop programs and their procedures. Procedural programming focuses on procedures or actions that are carried out in the program. Object-oriented programming focuses on objects and their attributes and behaviors. | | | *POINTS:* | 1 | | *REFERENCES:* | 27 Understanding the Evolution of Programming Models | | *QUESTION TYPE:* | True / False | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.07 - Describe the evolution of programming models | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11. \_\_\_\_ data items may involve organizing or sorting them, checking them for accuracy, or performing calculations with them.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | Processing | b. | Inputting | |  | c. | Outputting | d. | Converting |  |  |  | | --- | --- | | *ANSWER:* | a | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Correct. Processing data, which is done by the CPU, includes operations such as sorting times, checking items for accuracy, or performing calculations. | |  | b. | Incorrect. Inputting data consists of entering data items into a computer and storing them in memory, where they can be processed. | |  | c. | Incorrect. Outputting data occurs after data has been processed, when the resulting information is usually sent to a printer, monitor, or some other output device so that people can view, interpret, and use the results. | |  | d. | Incorrect. Converting data means to change its format. A compiler or interpreter converts syntax to machine language. | | | *POINTS:* | 1 | | *REFERENCES:* | 2 Understanding Computer Systems | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.01 - Describe computer systems | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 12. \_\_\_\_ errors are relatively easy to locate and correct because the compiler or interpreter you use highlights every error.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | Logic | b. | Syntax | |  | c. | Input | d. | Process |  |  |  | | --- | --- | | *ANSWER:* | b | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. Logic errors are due to incorrect instructions or sequences in programs. Compilers and interpreters can determine if a programmer used proper syntax but cannot determine if the instructions or sequences perform the intended output. Logical errors produce incorrect output. | |  | b. | Correct. Every language has rules governing its word usage and punctuation. These rules are called the language’s syntax. Because compilers and interpreters know the proper syntax, they can highlight incorrect text. | |  | c. | Incorrect. Input errors occur at runtime, which for compiled programs occurs when the CPU is executing a program that has already been converted into machine language. The compiler identifies incorrect syntax before compiling the program. | |  | d. | Incorrect. "Process error" is a generic term describing an error that occurs somewhere in a process. The process may be a debugging process or a mathematical process, so a process that gets interrupted unexpectedly experiences a process error. | | | *POINTS:* | 1 | | *REFERENCES:* | 4 Understanding Computer Systems | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.01 - Describe computer systems | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13. A \_\_\_\_ error results when you use a syntactically correct statement but use the wrong one for the current context.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | syntax | b. | logical | |  | c. | semantic | d. | line |  |  |  | | --- | --- | | *ANSWER:* | b | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. A syntax error occurs when you use the wrong word or punctuation defined by the programming language. | |  | b. | Correct. A logical error occurs when the syntax is correct but the instruction or sequence programmed is incorrect. | |  | c. | Incorrect. In programming, semantics describes the meaning of the syntax of any given statement. | |  | d. | Incorrect. A "line error" is an undefined term, but describes a line of code that includes a syntax or logical error. | | | *POINTS:* | 1 | | *REFERENCES:* | 12 Testing the Program | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 14. A(n) \_\_\_\_ is a named memory location whose value can vary.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | input | b. | variable | |  | c. | output | d. | logic |  |  |  | | --- | --- | | *ANSWER:* | b | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. Input is the action of entering data into a program. | |  | b. | Correct. Variables are named memory locations that can hold different values at different points in time. | |  | c. | Incorrect. Output is the action of sending resulting information to a printer, monitor, or some other device so that people can view, interpret, and use the results. | |  | d. | Incorrect. Logic is sequential instruction. | | | *POINTS:* | 1 | | *REFERENCES:* | 6 Understanding Simple Program Logic | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.02 - Understand simple program logic | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15. Computer programmers often refer to memory addresses using \_\_\_\_ notation.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | binary | b. | indirect | |  | c. | mathematical | d. | hexadecimal |  |  |  | | --- | --- | | *ANSWER:* | d | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. Binary is a two-digit numbering system used in machine language. Because large numbers represented in binary contain many digits, it is not a good numbering system for programmers to use to refer to memory addresses. | |  | b. | Incorrect. Indirect notation refers to the programming concept used in certain languages where the method precedes the object. | |  | c. | Incorrect. Mathematical notation deals with symbols and operands used to describe mathematical or logical operations. | |  | d. | Correct. Hexadecimal notation is a base 16 number system that is used to represent large binary numbers more compactly. | | | *POINTS:* | 1 | | *REFERENCES:* | 7 Understanding Simple Program Logic | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.02 - Understand simple program logic | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16. Before a programmer plans the logic of the program, he or she must \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | understand the problem | b. | write the program code | |  | c. | test the program | d. | maintain the program |  |  |  | | --- | --- | | *ANSWER:* | a | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Correct. The first step of the program development cycle is to understand the problem. | |  | b. | Incorrect. The third step of the program development cycle is to write the program code. | |  | c. | Incorrect. The fifth step of the program development cycle is to test the program. | |  | d. | Incorrect. The seventh, or final, step of the program development cycle is to maintain the program. | | | *POINTS:* | 1 | | *REFERENCES:* | 8 Understanding the Program Development Cycle | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 17. After a programmer plans the logic of a program, the next step is \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | understanding the problem | b. | testing the program | |  | c. | translating the program | d. | coding the program |  |  |  | | --- | --- | | *ANSWER:* | d | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. Understanding the problem is the first step of the program development cycle. | |  | b. | Incorrect. Testing the program is the fifth step of the program development cycle. | |  | c. | Incorrect. Translating the program is the fourth step of the program development cycle and is done through software, such as a compiler or interpreter. | |  | d. | Correct. Coding the program is the third step of the program development cycle and is completed after planning the logic and before translating the program into machine language. | | | *POINTS:* | 1 | | *REFERENCES:* | 10 Planning the Logic | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 18. The process of walking through a program’s logic on paper before you actually write the program is called \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | desk-checking | b. | flowcharting | |  | c. | pseudocoding | d. | testing |  |  |  | | --- | --- | | *ANSWER:* | a | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Correct. Planning the logic includes thinking carefully about all the possible data values a program might encounter and how you want the program to handle each scenario. Desk-checking is walking through this process on paper. | |  | b. | Incorrect. Flowcharting is a tool used while planning the logic of a program. | |  | c. | Incorrect. Pseudocoding is a tool used while planning the logic of a program. | |  | d. | Incorrect. Testing is the fifth step of the program development cycle and occurs after a compiler or interpreter translates the program into machine language. | | | *POINTS:* | 1 | | *REFERENCES:* | 10 Planning the Logic | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 19. Typically, a programmer develops a program’s logic, writes the code, and \_\_\_\_ the program, receiving a list of syntax errors.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | runs | b. | compiles | |  | c. | executes | d. | tests |  |  |  | | --- | --- | | *ANSWER:* | b | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. The program will not be run by the user until it has been compiled and checked for syntax errors. | |  | b. | Correct. The compiler will check the code for syntax errors before it translates the program into machine language. | |  | c. | Incorrect. The program will not be executed by the CPU until it has been compiled and checked for syntax errors. | |  | d. | Incorrect. Testing occurs after the program is successfully compiled. | | | *POINTS:* | 1 | | *REFERENCES:* | 11 Using Software to Translate the Program into Machine Language | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20. The process of finding and correcting program errors is called \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | desk-checking | b. | syntax-checking | |  | c. | compiling | d. | debugging |  |  |  | | --- | --- | | *ANSWER:* | d | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. The process of walking through a program’s logic on paper before you actually write the program is called desk-checking. | |  | b. | Incorrect. Syntax-checking is an automatic feature of most compilers. Programmers should review their syntax as they write code and use the automatic checking features of compilers. | |  | c. | Incorrect. Compiling is the process of translating high-level language statements into machine code. | |  | d. | Correct. Debugging is the process of finding and correcting program errors. | | | *POINTS:* | 1 | | *REFERENCES:* | 13 Testing the Program | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21. After programs are put into production, making necessary changes is called \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | debugging | b. | conversion | |  | c. | maintenance | d. | testing |  |  |  | | --- | --- | | *ANSWER:* | c | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. Debugging is the process of finding and correcting program errors. Debugging occurs during the testing phase of the program development cycle, which occurs before the program is put into production. | |  | b. | Incorrect. Conversion is the entire set of actions an organization must take to switch over to using a new program or set of programs. | |  | c. | Correct. Maintaining the program is the phase of the program development cycle that occurs after the program is put into production. | |  | d. | Incorrect. Testing is a phase of the program development cycle that deals with testing compiled code for logic errors. It involves debugging, which is the process of finding and correcting program errors. | | | *POINTS:* | 1 | | *REFERENCES:* | 14 Maintaining the Program | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 22. Using \_\_\_\_ involves writing down all the steps you will use in a program.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | a compiler | b. | an interpreter | |  | c. | desk-checking | d. | pseudocode |  |  |  | | --- | --- | | *ANSWER:* | d | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. A compiler is a piece of software that checks your code for syntax errors and translates the syntax into an executable file in machine language. | |  | b. | Incorrect. An interpreter is a piece of software that checks your code for syntax errors and translates the syntax into an executable file in machine language. | |  | c. | Incorrect. Desk-checking is the process of walking through a program on paper. | |  | d. | Correct. Pseudocode represents the logic of a program written out in English-like steps that appear similar to programming language. | | | *POINTS:* | 1 | | *REFERENCES:* | 15 Using Pseudocode Statements and Flowchart Symbols | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.04 - Write pseudocode statements and draw flowchart symbols | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 23. In a flowchart, the \_\_\_\_ is used to represent processing.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | lozenge | b. | diamond | |  | c. | parallelogram | d. | rectangle |  |  |  | | --- | --- | | *ANSWER:* | d | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. A lozenge represents a terminal, meaning the start and stop that occurs at each end of a flowchart. | |  | b. | Incorrect. A diamond represents a decision. | |  | c. | Incorrect. A triangle is not typically used. | |  | d. | Correct. A rectangle represents a process. | | | *POINTS:* | 1 | | *REFERENCES:* | 17 Drawing Flowcharts | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.04 - Write pseudocode statements and draw flowchart symbols | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 24. The \_\_\_\_ is used to represent output in a flowchart.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | square | b. | lozenge | |  | c. | parallelogram | d. | triangle |  |  |  | | --- | --- | | *ANSWER:* | c | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. A square is not typically used in a flowchart. | |  | b. | Incorrect. A lozenge represents a terminal, meaning the start and stop that occurs at each end of a flowchart. | |  | c. | Correct. A parallelogram is used to represent input or output. | |  | d. | Incorrect. A triangle is not typically used in a flowchart. | | | *POINTS:* | 1 | | *REFERENCES:* | 17 Drawing Flowcharts | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.04 - Write pseudocode statements and draw flowchart symbols | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 25. The \_\_\_\_ is the standard terminal symbol for a flowchart.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | parallelogram | b. | lozenge | |  | c. | diamond | d. | square |  |  |  | | --- | --- | | *ANSWER:* | b | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. A parallelogram is used to represent input or output. | |  | b. | Correct. A lozenge represents a terminal, meaning the start and stop that occurs at each end of a flowchart. | |  | c. | Incorrect. A diamond represents a decision. | |  | d. | Incorrect. A square is not typically used in a flowchart. | | | *POINTS:* | 1 | | *REFERENCES:* | 17 Drawing Flowcharts | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.04 - Write pseudocode statements and draw flowchart symbols | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 26. The repetition of a series of steps is called a(n) \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | repeat flow | b. | loop | |  | c. | flow | d. | algorithm |  |  |  | | --- | --- | | *ANSWER:* | b | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. Repeat flow refers to a loop control statement regarding the repetition of a process until a condition or stopping point is reached. | |  | b. | Correct. The repetition of a series of steps is called a loop. | |  | c. | Incorrect. *Flow* is a word that is used to describe continuous movement, such as the flow of data. | |  | d. | Incorrect. An algorithm is the sequence of steps or rules you follow to solve a problem. | | | *POINTS:* | 1 | | *REFERENCES:* | 19 Repeating Instructions | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.04 - Write pseudocode statements and draw flowchart symbols | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 27. A(n) \_\_\_\_ is a repeating flow of logic with no end.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | algorithm | b. | infinite loop | |  | c. | variable | d. | decision symbol |  |  |  | | --- | --- | | *ANSWER:* | b | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. An algorithm is the sequence of steps or rules you follow to solve a problem. | |  | b. | Correct. An infinite loop is a special circumstance in which a loop continues to repeat indefinitely. Infinite loops are usually due to an error in the loop control variable. | |  | c. | Incorrect. A variable is a named memory location whose value can hold different values at different points in time. | |  | d. | Incorrect. A diamond, or decision symbol, is a symbol used in flowcharting to represent testing a value or evaluating an expression. The diamond usually contains a question or evaluation. | | | *POINTS:* | 1 | | *REFERENCES:* | 19 Repeating Instructions | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.04 - Write pseudocode statements and draw flowchart symbols | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 28. You represent a decision in a flowchart by drawing a decision symbol, which is shaped like a \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | parallelogram | b. | square | |  | c. | diamond | d. | circle |  |  |  | | --- | --- | | *ANSWER:* | c | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. Parallelograms represent input and output operations. Decisions are represented by diamonds. | |  | b. | Incorrect. Squares are not typically used, but rectangles represent processing. Decisions are represented by diamonds. | |  | c. | Correct. The diamond, or decision symbol, is a symbol used in flowcharting to represent testing a value or evaluating an expression. The diamond usually contains a question or evaluation. | |  | d. | Incorrect. Circles sometimes used as connectors in a flowchart, but lozenges represent terminals (ends). Decisions are represented by diamonds. | | | *POINTS:* | 1 | | *REFERENCES:* | 21 Using a Sentinel Value to End a Program | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.05 - Use a sentinel value to end a program | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 29. Many programming languages use the term \_\_\_\_ to refer to the marker that is used to automatically recognize the end of data in a file.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | data\_end | b. | end\_data | |  | c. | eof | d. | eod |  |  |  | | --- | --- | | *ANSWER:* | c | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. *data\_end* is not a typical sentinel value used to represent the end of a file. | |  | b. | Incorrect. *end\_data* is not a typical sentinel value used to represent the end of a file. | |  | c. | Correct. *eof* is a typical sentinel value and represents the phrase *end of file*. | |  | d. | Incorrect. *eod* is not a typical sentinel value used to represent the end of a file. | | | *POINTS:* | 1 | | *REFERENCES:* | 22 Using a Sentinel Value to End a Program | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.05 - Use a sentinel value to end a program | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 30. To enter the program into a computer so you can translate and execute it, you usually use a keyboard to type program statements into a(n)\_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | command line | b. | input screen | |  | c. | editor | d. | word processor |  |  |  | | --- | --- | | *ANSWER:* | c | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. A command line is a location on a computer screen at which you type text to communicate with the computer’s operating system. | |  | b. | Incorrect. An input screen is a screen in a graphical user interface (GUI) where a user inputs data. | |  | c. | Correct. A programmer enters his or her program in the editor part of an integrated development environment (IDE). | |  | d. | Incorrect. A word processor is an application that allows you to enter text and then edit it. Word processors are similar to text editors but generally contain more features. | | | *POINTS:* | 1 | | *REFERENCES:* | 23 Understanding Programming Environments | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.06 - Understand programming and user environments | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31. A(n) \_\_\_\_ is a program that you use to create simple text files.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | text editor | b. | IDE | |  | c. | GUI | d. | GDE |  |  |  | | --- | --- | | *ANSWER:* | a | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Correct. A text editor is a program used to create simple text files; it is similar to a word processor but without as many features. | |  | b. | Incorrect. An integrated development environment (IDE) is a software package that provides an editor, compiler, and other programming tools. | |  | c. | Incorrect. A graphical user interface (GUI) allows users to interact with a program in a graphical environment. | |  | d. | Incorrect. The acronym GDE does not have a defined meaning when discussing logic and design; it is likely a confusion of the acronyms GUI and IDE. | | | *POINTS:* | 1 | | *REFERENCES:* | 23 Understanding Programming Environments | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.06 - Understand programming and user environments | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 32. A(n) \_\_\_\_ is a software package that provides an editor, a compiler, and other programming tools.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | GUI | b. | CGI | |  | c. | GDE | d. | IDE |  |  |  | | --- | --- | | *ANSWER:* | d | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. A graphical user interface (GUI) allows users to interact with a program in a graphical environment. | |  | b. | Incorrect. The acronym CGI does not have a defined meaning when discussing logic and design; it is generally used to represent computer-generated imagery, which is an application of computer graphics. | |  | c. | Incorrect. The acronym GDE does not have a defined meaning when discussing logic and design; it is likely a confusion of the acronyms GUI and IDE. | |  | d. | Correct. An integrated development environment (IDE) is a software package that provides an editor, compiler, and other programming tools. | | | *POINTS:* | 1 | | *REFERENCES:* | 24 Understanding Programming Environments | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.06 - Understand programming and user environments | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 33. A(n) \_\_\_\_ is a location on your computer screen where you type text entries to communicate with the computer’s operating system.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | input line | b. | communication line | |  | c. | command line | d. | GCI |  |  |  | | --- | --- | | *ANSWER:* | c | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. "Input line" is a colloquial term used to describe a line of input. | |  | b. | Incorrect. A communication line is a line or wire carrying communication. | |  | c. | Correct. A command line is a location on your screen used to communicate with the computer’s operating system. | |  | d. | Incorrect. The acronym *GCI* does not have a defined meaning when discussing logic and design. | | | *POINTS:* | 1 | | *REFERENCES:* | 25 Understanding User Environments | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.06 - Understand programming and user environments | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 34. A \_\_\_\_ allows users to interact with a program in a graphical environment.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | GCI | b. | CGI | |  | c. | command line | d. | GUI |  |  |  | | --- | --- | | *ANSWER:* | d | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. The acronym GCI does not have a defined meaning when discussing logic and design. | |  | b. | Incorrect. The acronym CGI does not have a defined meaning when discussing logic and design; it is generally used to represent computer-generated imagery, which is an application of computer graphics. | |  | c. | Incorrect. A command line is a location on your screen used to communicate with the computer’s operating system. | |  | d. | Correct. A graphical user interface (GUI) allows users to interact with a program in a graphical environment. | | | *POINTS:* | 1 | | *REFERENCES:* | 25 Understanding User Environments | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.06 - Understand programming and user environments | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 35. The major difference between the two main programming styles in use today is the \_\_\_\_.   |  |  |  | | --- | --- | --- | |  | a. | use of flowcharts versus pseudocode | |  | b. | testing procedure used by the programmer | |  | c. | programmer’s focus during the earliest planning stages of a project | |  | d. | programming language used |  |  |  | | --- | --- | | *ANSWER:* | c | | *FEEDBACK:* | |  |  |  | | --- | --- | --- | |  | a. | Incorrect. Flowcharts and pseudocode are tools used to plan logic. | |  | b. | Incorrect. The testing procedure is part of a program design cycle and is not considered a programming style. | |  | c. | Correct. The programmer’s focus during the earliest planning stages may be procedural or object-oriented. | |  | d. | Incorrect. The language that the programmer uses is not considered a programming style. However, different languages are designed and better suited for one style over another. | | | *POINTS:* | 1 | | *REFERENCES:* | 27 Understanding the Evolution of Programming Models | | *QUESTION TYPE:* | Multiple Choice | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.07 - Describe the evolution of programming models | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 36. When instructions are carried out, a program runs, or \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.   |  |  | | --- | --- | | *ANSWER:* | executes | | *POINTS:* | 1 | | *REFERENCES:* | 4 Understanding Computer Systems | | *QUESTION TYPE:* | Completion | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.01 - Describe computer systems | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 37. A(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is the sequence of steps necessary to solve any problem.   |  |  | | --- | --- | | *ANSWER:* | algorithm | | *POINTS:* | 1 | | *REFERENCES:* | 10 Planning the Logic | | *QUESTION TYPE:* | Completion | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 38. When you draw a flowchart, you use a(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ to represent the input symbol, which indicates an input operation.   |  |  | | --- | --- | | *ANSWER:* | parallelogram | | *POINTS:* | 1 | | *REFERENCES:* | 17 Drawing Flowcharts | | *QUESTION TYPE:* | Completion | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.04 - Write pseudocode statements and draw flowchart symbols | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 39. A preselected value that stops the execution of a program is often called a(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ value because it does not represent real data.   |  |  | | --- | --- | | *ANSWER:* | dummy | | *POINTS:* | 1 | | *REFERENCES:* | 22 Using a Sentinel Value to End a Program | | *QUESTION TYPE:* | Completion | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.05 - Use a sentinel value to end a program | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 40. A(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ programmer would approach a problem by dividing the process into manageable subtasks.   |  |  | | --- | --- | | *ANSWER:* | procedural | | *POINTS:* | 1 | | *REFERENCES:* | 27 Understanding the Evolution of Programming Models | | *QUESTION TYPE:* | Completion | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.07 - Describe the evolution of programming models | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Match each term with a statement below.*   |  |  | | --- | --- | | a. | volatile memory | | b. | software | | c. | syntax error | | d. | machine language | | e. | sentinel | | f. | pseudocode | | g. | hardware | | h. | flowchart | | i. | flowlines | | j. | documentation |  |  |  | | --- | --- | | *REFERENCES:* | 15, Using Pseudocode Statements and Flowchart Symbols 17 2 22 4 9 | | *QUESTION TYPE:* | Matching | | *HAS VARIABLES:* | False | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.01 - Describe computer systems PL&D.FARR.18.01.03 - List the steps involved in the program development cycle PL&D.FARR.18.01.04 - Write pseudocode statements and draw flowchart symbols PL&D.FARR.18.01.05 - Use a sentinel value to end a program | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 41. Equipment, or the physical devices, associated with a computer   |  |  | | --- | --- | | *ANSWER:* | g | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 42. Instructions that tell the computer what to do   |  |  | | --- | --- | | *ANSWER:* | b | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 43. Contents are lost when the computer is turned off or loses power   |  |  | | --- | --- | | *ANSWER:* | a | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 44. Represent(s) the millions of on/off circuits within the computer   |  |  | | --- | --- | | *ANSWER:* | d | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 45. Incorrectly spelled words, or reversing the proper order of two words in a computer program   |  |  | | --- | --- | | *ANSWER:* | c | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 46. All the supporting paperwork for a program   |  |  | | --- | --- | | *ANSWER:* | j | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 47. Pictorial representation of the logical steps it takes to solve a problem   |  |  | | --- | --- | | *ANSWER:* | h | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 48. English-like representation of the logical steps it takes to solve a problem   |  |  | | --- | --- | | *ANSWER:* | f | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 49. Used to show the correct sequence of statements   |  |  | | --- | --- | | *ANSWER:* | i | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 50. Preselected value that stops the execution of a program   |  |  | | --- | --- | | *ANSWER:* | e | | *POINTS:* | 1 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 51. Describe temporary and permanent storage.   |  |  | | --- | --- | | *ANSWER:* | When you type program instructions, they are stored in computer memory, which is a computer’s temporary, internal storage. Internal storage is volatile—its contents are lost when the computer is turned off or loses power. Usually, you want to be able to retrieve and perhaps modify the stored instructions later, so you also store them on a permanent storage device, such as a disk. Permanent storage devices are nonvolatile—that is, their contents are persistent and are retained even when power is lost. | | *POINTS:* | 1 | | *REFERENCES:* | 4 Understanding Computer Systems | | *QUESTION TYPE:* | Subjective Short Answer | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.01 - Describe computer systems | | *TOPICS:* | Critical Thinking | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 52. Describe the function of a compiler and an interpreter.   |  |  | | --- | --- | | *ANSWER:* | Each programming language uses a piece of software, called a compiler or an interpreter, to translate your program code into machine language. Machine language is also called binary language and is represented as a series of 0s and 1s. The compiler or interpreter that translates your code tells you if any programming language component has been used incorrectly. Syntax errors are relatively easy to locate and correct because the compiler or interpreter you use highlights every syntax error. If you write a computer program using a language such as C++ but spell one of its words incorrectly or reverse the proper order of two words, the software lets you know that it found a mistake by displaying an error message as soon as you try to translate the program. | | *POINTS:* | 1 | | *REFERENCES:* | 4 Understanding Computer Systems | | *QUESTION TYPE:* | Subjective Short Answer | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.01 - Describe computer systems | | *TOPICS:* | Critical Thinking | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 53. What is a variable and how is it used in a program?   |  |  | | --- | --- | | *ANSWER:* | Suppose the location myNumber is a variable. A variable is a named memory location whose value can vary—for example, the value of myNumber might be 3 when the program is used for the first time, and 45 when it is used the next time. | | *POINTS:* | 1 | | *REFERENCES:* | 6 Understanding Simple Program Logic | | *QUESTION TYPE:* | Subjective Short Answer | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.02 - Understand simple program logic | | *TOPICS:* | Critical Thinking | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 54. What is involved in putting a program into production?   |  |  | | --- | --- | | *ANSWER:* | Once the program is tested adequately, it is ready for the organization to use. Putting the program into production might mean simply running the program once, if it was written to satisfy a user’s request for a special list. However, the process might take months if the program will be run on a regular basis, or if it is one of a large system of programs being developed. Perhaps data-entry people must be trained to prepare the input for the new program; users must be trained to understand the output; or existing data in the company must be changed to an entirely new format to accommodate this program. Conversion, the entire set of actions an organization must take to switch over to using a new program or set of programs, can sometimes take months or years to accomplish. | | *POINTS:* | 1 | | *REFERENCES:* | 13-14 Putting the Program into Production | | *QUESTION TYPE:* | Subjective Short Answer | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.03 - List the steps involved in the program development cycle | | *TOPICS:* | Critical Thinking | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 55. How is a decision represented in a flowchart?   |  |  | | --- | --- | | *ANSWER:* | You represent a decision in a flowchart by drawing a decision symbol, which is shaped like a diamond. The diamond usually contains a question, the answer to which is one of two mutually exclusive options—often yes or no. All good computer questions have only two mutually exclusive answers, such as yes and no, or true and false. For example, “What day of the year is your birthday?” is not a good computer question because there are 366 possible answers. But “Is your birthday June 24?” is a good computer question because, for everyone in the world, the answer is either yes or no. | | *POINTS:* | 1 | | *REFERENCES:* | 21 Using a Sentinel Value to End a Program | | *QUESTION TYPE:* | Subjective Short Answer | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.05 - Use a sentinel value to end a program | | *TOPICS:* | Critical Thinking | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 56. How does a program recognize the end of an input file?   |  |  | | --- | --- | | *ANSWER:* | Programming languages can recognize the end of data in a file automatically, through a code that is stored at the end of the data. Many programming languages use the term eof (for “end of file”) to refer to this marker that automatically acts as a sentinel. | | *POINTS:* | 1 | | *REFERENCES:* | 22 Using a Sentinel Value to End a Program | | *QUESTION TYPE:* | Subjective Short Answer | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.05 - Use a sentinel value to end a program | | *TOPICS:* | Critical Thinking | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 57. Discuss two ways that you can enter a program into a computer.   |  |  | | --- | --- | | *ANSWER:* | To enter the program into a computer so you can translate and execute it, you usually use a keyboard to type program statements into an editor. You can type a program into one of the following:  A plain text editor  A text editor that is part of an integrated development environment  A text editor is a program that you use to create simple text files. It is similar to a word processor, but without as many features. You can use a text editor such as Notepad that is included with Microsoft Windows. An advantage to using a simple text editor to type and save a program is that the completed program does not require much disk space for storage.  You can use the editor of an integrated development environment (IDE) to enter your program. An IDE is a software package that provides an editor, a compiler, and other programming tools. | | *POINTS:* | 1 | | *REFERENCES:* | 23-24 Understanding Programming Environments | | *QUESTION TYPE:* | Subjective Short Answer | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.06 - Understand programming and user environments | | *TOPICS:* | Critical Thinking | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 58. List the features that are commonly included with an IDE editor.   |  |  | | --- | --- | | *ANSWER:* | It uses different colors to display various language components, making elements like data types easier to identify.  It highlights syntax errors visually for you.  It employs automatic statement completion; when you start to type a statement, the IDE suggests a likely completion, which you can accept with a keystroke.  It provides tools that allow you to step through a program’s execution one statement at a time so you can more easily follow the program’s logic and determine the source of any errors. | | *POINTS:* | 1 | | *REFERENCES:* | 25 Understanding Programming Environments | | *QUESTION TYPE:* | Subjective Short Answer | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.06 - Understand programming and user environments | | *TOPICS:* | Critical Thinking | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 59. What are some of the features provided by modern programming languages that were not available in older programming languages?   |  |  | | --- | --- | | *ANSWER:* | People have been writing modern computer programs since the 1940s. The oldest programming languages required programmers to work with memory addresses and to memorize awkward codes associated with machine languages. Newer programming languages look much more like natural language and are easier to use, partly because they allow programmers to name variables instead of using awkward memory addresses. Also, newer programming languages allow programmers to create self-contained modules or program segments that can be pieced together in a variety of ways. The oldest computer programs were written in one piece, from start to finish, but modern programs are rarely written that way—they are created by teams of programmers, each developing reusable and connectable program procedures. Writing several small modules is easier than writing one large program, and most large tasks are easier when you break the work into units and get other workers to help with some of the units. | | *POINTS:* | 1 | | *REFERENCES:* | 27 Understanding the Evolution of Programming Models | | *QUESTION TYPE:* | Subjective Short Answer | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.07 - Describe the evolution of programming models | | *TOPICS:* | Critical Thinking | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 60. Currently, there are two major techniques used to develop programs and their procedures. Name and describe them.   |  |  | | --- | --- | | *ANSWER:* | One technique, procedural programming, focuses on the procedures that programmers create. In other words, procedural programmers focus on the actions that are carried out—for example, getting input data for an employee and writing the calculations needed to produce a paycheck from the data. Procedural programmers would approach the job of producing a paycheck by breaking down the paycheck-producing process into manageable subtasks.  The other popular programming model, object-oriented programming, focuses on objects, or “things,” and describes their features (or attributes) and their behaviors. For example, object-oriented programmers might design a payroll application by thinking about employees and paychecks, and describing their attributes (e.g., employees have names and Social Security numbers, and paychecks have names and check amounts). | | *POINTS:* | 1 | | *REFERENCES:* | 27 Understanding the Evolution of Programming Models | | *QUESTION TYPE:* | Subjective Short Answer | | *HAS VARIABLES:* | False | | *STUDENT ENTRY MODE:* | Basic | | *LEARNING OBJECTIVES:* | PL&D.FARR.18.01.07 - Describe the evolution of programming models | | *TOPICS:* | Critical Thinking | | *DATE CREATED:* | 9/12/2016 12:40 PM | | *DATE MODIFIED:* | 11/15/2019 4:09 PM | |