CIS170 Final Notes:

Week 1

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1. | Question : | (TCO 2) Which characters make up the Stream insertion operator? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | :) | |  |  |  | :: | |  |  | CORRECT | << | |  |  |  | // | |  | Instructor Explanation: | Chapter 1, p. 11: The << operator is referred to as the Stream insertion operator. | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 2. | Question : | (TCO 2) In C++, what is an example of a header file? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Iterations | |  |  |  | Variables | |  |  | CORRECT | iostream | |  |  |  | cout | |  | Instructor Explanation: | Chapter 1, p. 11: iostream is called a header file because it is at the head of your program. | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 3. | Question : | (TCO 2) The body of the main function starts with a \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | [ | |  |  | CORRECT | { | |  |  |  | } | |  |  |  | ] | |  | Instructor Explanation: | Chapter 1, p. 11: It starts with a left brace {. | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 4. | Question : | (TCO 2) C++ comments are represented by which characters? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | // and /\* \*/ | |  |  |  | :: and // | |  |  |  | // and >> | |  |  |  | > | |  | Instructor Explanation: | Chapter 1, p. 11 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 5. | Question : | (TCO 2) A Main Function is enclosed inside \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | box brackets | |  |  | CORRECT | curly braces | |  |  |  | parentheses | |  |  |  | quotes | |  | Instructor Explanation: | Chapter 1, p. 11 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 6. | Question : | (TCO 1) Microsoft Visual C++ is an example of a(n) \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | C++ programming language | |  |  |  | debugger | |  |  |  | compiler | |  |  | CORRECT | IDE | |  | Instructor Explanation: | Chapter 1, p. 13 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 7. | Question : | (TCO 2) Which of the following statements will output “Hello World”? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | cout | |  |  |  | cin | |  |  |  | Print Queue | |  |  |  | Print Line | |  | Instructor Explanation: | Chapter 1: Multiple code examples sending statements to the console with cout | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 8. | Question : | (TCO 2) To assign a value 1 to variable x, you write \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | x++; | |  |  |  | 1 = x; | |  |  | CORRECT | x = 1; | |  |  |  | x == 1; | |  | Instructor Explanation: | Chapter 2, p. 40 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 9. | Question : | (TCO 3) What is the value of x after the following statement? float x;  x = 2 \* 5 + (3 + 1)/ 5.0 ; | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | 10.0 | |  |  | CORRECT | 10.8 | |  |  |  | 1.0 | |  |  |  | 0.8 | |  | Instructor Explanation: | Chapter 2, p. 43 on data type: Float; see p. 46 for Arithmetic Expressions | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 10. | Question : | (TCO 2) From the list below, which is an INVALID variable name? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | CIS\_170 | |  |  |  | \_CIS | |  |  | CORRECT | 2\_More | |  |  |  | twoMore | |  | Instructor Explanation: | Chapter 2, p. 38, Section 2.4 on Identifiers: A variable cannot start with a digit. | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 11. | Question : | (TCO 3) What is the output of the following program?  #include <iostream> using namespace std;  int main() {     int x1, x2, i, j, k, y, z;     float f;     x1 = 1;     x2 = 1;     y = 5 + x1--;     z = 5 + ++x2;     i = 6 % 4;     j = 1;     j += j + 3;     k = 25 / 2;     f = (float)((2 / 5) \* k);      cout << "x1 is " << x1 << endl;     cout << "x2 is " << x2 << endl;     cout << " i is " << i << endl;     cout << " j is " << j << endl;     cout << " k is " << k << endl;     cout << " y is " << y << endl;     cout << " z is " << z << endl;     cout << " f is " << f << endl;     return 0; } | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | x1 is 0 x2 is 2 i is 2 j is 5 k is 12 y is 6 z is 7 f is 0 x1 equals 0 because x1 decrements as part of y. so it was 1 then decrement it and it's 0. x2 equals 2 because it increments as part of z, so it was 1 then it increments so becomes 2. i is 2 because 6 modulos 4 is 2 j is 5 because if you consider j = 1 then you increment it, it becomes 2, then you add 3 and that gives you 5. k equals 12 because if you divide 25 by 2 you get 12.5. Since this is an integer and not a floating point we take off the .5 and it equals 12. f equals 0 because (float) times 4.8 equals 0. | |  | Instructor Explanation: | Lecture and Chapter 2  x1 is 0  x2 is 2    i is 2    j is 5    k is 12    y is 6    z is 7    f is 0 | | |

Week2

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1. | Question : | (TCO 4) \_\_\_\_\_ is often used in the design of a program to describe the program processing. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | Pseudocode | |  |  |  | Algorithm | |  |  |  | Program control | |  |  |  | Data flow | |  | Instructor Explanation: | Lecture / Chapter 3, B.1 Introduction | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 2. | Question : | (TCO 4) The symbol "==" is used as part of a \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | comparison operator | |  |  |  | decision operator | |  |  |  | repetition operator | |  |  |  | sequence operator | |  | Instructor Explanation: | Chapter 3, Section 3.2: The Bool Data Type | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 3. | Question : | (TCO 4) Which of the following is true about if/else structure? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | The if statement contains three parts. The first part is a keyword that tells the computer that it's an if statement. The second part is a boolean expression. The third part is a statement or group of statements. | |  |  |  | You must always surround a single statement after an if with curly braces. | |  |  |  | You must always match an if with an else. | |  |  |  | An else must always be associated with an if, and vice versa. | |  | Instructor Explanation: | Lecture / Chapter 3 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 4. | Question : | (TCO 4) Which one of these symbols is a logical operator? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | || | |  |  |  | III | |  |  |  | IIII | |  |  |  | :: | |  | Instructor Explanation: | Chapter 3, Section 3.11: Logical Operators | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 5. | Question : | (TCO 4) How many Relational Operator symbols are there? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Four | |  |  |  | Five | |  |  | CORRECT | Six | |  |  |  | Eight | |  | Instructor Explanation: | Chapter 3, Section 3.2 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 6. | Question : | (TCO 4) The \_\_\_\_\_ of a variable is limited to the block in which it is declared. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | branching ability | |  |  | CORRECT | scope | |  |  |  | associativity | |  |  |  | precedence | |  | Instructor Explanation: | Lecture / Chapter 3 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 7. | Question : | (TCO 4) Input values should always be checked for \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | appropriate range | |  |  |  | reasonableness | |  |  |  | division by zero if division is taking place | |  |  | CORRECT | All of the above | |  | Instructor Explanation: | Lecture / Chapter 3 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 8. | Question : | (TCO 4) Consider the following segment of code:  if(apple == 5)       cout<<"You got \"five\" apples!"<<endl;    else       cout<<"You do not have five apples!\n";    cout<<"The end of the program is reached.";  What will be the output on screen if apple is 4? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | You got five apples!    The end of the program is reached. | |  |  |  | You got five apples! | |  |  |  | You do not have five apples! | |  |  | CORRECT | You do not have five apples! The end of the program is reached. | |  | Instructor Explanation: | The programmer used relational operators, and thus, 5 is compared to the apple. Since the apple was four, it would not be true that apple = = 5, so you would not have five apples. The end of program line would print and is not dependent upon the if statement. | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 9. | Question : | (TCO 4) Case Statements are checked \_\_\_\_\_, but the order of the cases does not matter. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | according to their data value | |  |  | CORRECT | in sequential order | |  |  |  | in random order | |  |  |  | in break order and then default order according to the parent conditional expression | |  | Instructor Explanation: | Chapter 3, Section 3.14: switch statements (switch case rules) | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 10. | Question : | (TCO 6) If there are errors in the executable code and it needs to be run one step at a time to find the errors, what development tool should be used? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Compiler | |  |  | CORRECT | Debugger | |  |  |  | Linker | |  |  |  | Common language runtime | |  | Instructor Explanation: | Lecture / Chapter 3 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 11. | Question : | (TCO 4) What is the advantage of using the conditional operator? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | The advantage of using a conditional operator is that it uses less code and so you don't have to code out an entire if statement. You can use just one line instead of 8. It's also easier to read the code with less. | |  | Instructor Explanation: | (Lecture / Chapter 3) The advantage of using the conditional operator is the ability to place a decision and its two possible outcomes in an abbreviated format. | | |

Week3

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1. | Question : | (TCO 5) Looping statements follow the \_\_\_\_\_ structure. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | sequence | |  |  |  | control | |  |  |  | selection | |  |  | CORRECT | repetition | |  | Instructor Explanation: | Lecture / Chapter 4 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 2. | Question : | (TCO 5) Which repetition structure is designed to have the code execute at least one time? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | For loop | |  |  |  | Do in a while loop | |  |  |  | While loop | |  |  | CORRECT | Do-while loop | |  | Instructor Explanation: | Chapter 4, Section 4.3: do-while loop section | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 3. | Question : | (TCO 10) To halt the execution of a program at a specific spot during debugging, you must use a \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | scope resolution operator | |  |  |  | step | |  |  |  | comment | |  |  | CORRECT | break point | |  | Instructor Explanation: | Chapter 2, Section 2.14 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 4. | Question : | (TCO 5) The while loop is which type of loop? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Posttest loop | |  |  | CORRECT | Pretest loop | |  |  |  | Infinite loop | |  |  |  | None of the above | |  | Instructor Explanation: | Chapter 4, Section 4.2: the while loop section | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 5. | Question : | (TCO 5) A loop that never stops is called a(n) \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | nested loop | |  |  | CORRECT | infinite loop | |  |  |  | pretest loop | |  |  |  | posttest loop | |  | Instructor Explanation: | Chapter 4, Section 4.6: Nested Loops | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 6. | Question : | (TCO 5) Float and double variables should not be used \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | as counters | |  |  |  | to perform mathematical calculations | |  |  |  | as approximate representations of decimal numbers | |  |  |  | for applications where precision is required | |  | Instructor Explanation: | Chapter 4, Section 4.7 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 7. | Question : | (TCO 5) A loop that falls entirely within the body of another is a(n) \_\_\_\_ loop. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | outer | |  |  | CORRECT | nested | |  |  |  | parent | |  |  |  | restricted | |  | Instructor Explanation: | Chapter 4, Section 4.6: Nested section | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 8. | Question : | (TCO 5) How many times will this for loop execute?   for (int i = 100; I > 0; --i) | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Zero - the loop will not execute | |  |  |  | 99 | |  |  |  | 100 | |  |  | CORRECT | The loop is not written correctly. It would not compile. | |  | Instructor Explanation: | Chapter 4, Section 4.4 - C++ is case sensitive and "i" is written as "I" when it should not have been. | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 9. | Question : | (TCO 5) In the following segment of code, the body of the loop is \_\_\_\_\_.  number = 1; while(number <= 10); {      cout<<number<<endl;      ++number; } | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | everything in between the curly brackets | |  |  | CORRECT | empty | |  |  |  | cout | |  |  |  | ++number; | |  | Instructor Explanation: | Lecture / Chapter 4 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 10. | Question : | (TCO 5) Which of the following for loop is valid? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | INCORRECT | for(int i=1; i<=5; ++i) | |  |  |  | int i; for(i=1; i<=5; i++) | |  |  |  | int i=1; for(; i<=5; i++) | |  |  | CORRECT | All of the above | |  | Instructor Explanation: | Lecture / Chapter 4, Section 4.4 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 11. | Question : | (TCO 5) Using a loop, write a program that adds and displays the total of all even integers from 1 to 10. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | #include<iostream> using namespace std; int main(){ int total=0; int i; cout<<"All the even numbers from 1 - 10 are listed below:\n" << endl; for (i = 2; i<=10; i= i+2) { total = total + i; cout<<i<< endl; } cout << "\nThe total of the even numbers is: " << total << endl; cin.ignore(2); } | |  | Instructor Explanation: | Lecture / Chapter 4  #include <iostream>  using namespace std;  int main() {       int x, sum = 0;      for(int i =0; i <= 10; i+=2)           {                sum = sum + i;           }      cout << "sum: " << sum << endl;      return 0; } | | |

Week 4

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1. | Question : | (TCO 7) Code written in a specific format to perform a single task or calculation is called a \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | function | |  |  |  | loop | |  |  |  | decision statement | |  |  |  | task | |  | Instructor Explanation: | Chapter 5, Section 5.2 Defining a Function | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 2. | Question : | (TCO 8) Unless otherwise specified, C++ variables are automatically passed by \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | parameter | |  |  |  | argument | |  |  |  | reference | |  |  | CORRECT | value | |  | Instructor Explanation: | Lecture; Chapter 5, Section 5.2 Defining a Function (top of p. 157) | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 3. | Question : | (TCO 7) When is a return statement required inside a function? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Every function must include at least one return statement. | |  |  |  | When you use a switch statement | |  |  | CORRECT | When a return data type (other than void) is included in the function prototype | |  |  |  | Inside an if-else statement | |  | Instructor Explanation: | Lecture; Chapter 5, Section 5.2 Defining a Function | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 4. | Question : | (TCO 7) In this prototype, what is missing?  void PrintNumber(int ) | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Nothing is missing, it is complete. | |  |  | INCORRECT | The variable name must be included in the input list. | |  |  | CORRECT | A semicolon | |  |  |  | A return type | |  | Instructor Explanation: | Lecture; Chapter 5, Section 5.2 Defining a Function | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 5. | Question : | (TCO 7) What does it mean to say a variable is local to a function? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | main() can use it whenever it wants to, but no other function can see or use it. | |  |  | CORRECT | Only the function where the variable is declared can see and use it. | |  |  |  | No functions can see or use it. | |  |  |  | Every function can use it. | |  | Instructor Explanation: | Lecture; Chapter 6, Section 6.5 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 6. | Question : | (TCO 7) A variable declared inside of a for-loop body has its scope \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | within the variable | |  |  | INCORRECT | within the entire loop | |  |  | CORRECT | limited in the body of the loop | |  |  |  | limited to the statement line | |  | Instructor Explanation: | Chapter 6, Section 6.5.1 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 7. | Question : | (TCO 8) Which function prototype indicates the parameter is passed by reference? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | int myFunction( double &) | |  |  |  | int myFunction(double) | |  |  |  | int myFunction(double \*) | |  |  | INCORRECT | int myFunction(&) | |  | Instructor Explanation: | Lecture; Chapter 6, Section 6.4 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 8. | Question : | (TCO 8) What would be a valid argument for the following function?  char myFunction(int &x) | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | int myVariable; | |  |  |  | char myVariable; | |  |  |  | 8 | |  |  | INCORRECT | All of the above | |  | Instructor Explanation: | Lecture; Chapter 6, Section 6.1 Introduction | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 9. | Question : | (TCO 7) What is wrong with this code?  void PrinttoScreen( int dollars, int cents) {      cout << “\n The dollars are << dollars;      cout << “\n The cents are << cents;      return OK } | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | The return type is not void. | |  |  |  | You can't return an OK. | |  |  | CORRECT | The return type is void, AND you can't return an OK. | |  |  |  | Nothing is wrong with it. | |  | Instructor Explanation: | Lecture and Chapter 5 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 10. | Question : | (TCO 7) What is the purpose of the function call? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | Control is passed to the called function. | |  |  |  | The program is passed to all of the called function's variables. | |  |  |  | main() gets to use the function's variables. | |  |  |  | Control is passed to the called function AND main() gets to use the function's variables. | |  | Instructor Explanation: | Lecture and Chapter 5 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 11. | Question : | (TCO 7) In your own words, define a function and code an example of creating a function and calling it. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | A function is a separate piece of code that can be written one time, but called many times without having to re-write the code. An example of this might be: # include<iostream> # include<string> using namespace std; void name(); int main() { name(); cout<<"Welcome to the world of C++!"<<endl; cin.ignore(2); return 0; } void name() { string name; cout<< "What is your name? "; getline(cin, name); cout<< "\nHi "<<name<<", "; } | |  | Instructor Explanation: | A function is a construct for grouping statements together to perform a task. Using a function, you can write the code once to perform the task in a program and then reuse it in many other programs. For example, if you need to find the larger of two numbers, you could create a function like the following:    int maxNumberFunction (int num1, int num2) {      int result;      if (num1 > num2)           result = num1;      else           result = num2;      return result; }    If you define a function to perform this task, you don’t have to repeatedly write the same code. You define it just once and then reuse it on other programs.    One way to call this function would be:    cout << maxNumberFunction (2,5);    When you call or invoke it, you are now instructing it to be used, as defined it is merely created.    Thank you  Chapter 5, Sections 5.1/5.2/5.3 | | |

Week5

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1. | Question : | (TCO 11) If n represents the size of an array, what would be the index of the last element in the array? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | n-2 | |  |  |  | n | |  |  | CORRECT | n-1 | |  |  |  | Cannot be determined | |  | Instructor Explanation: | Lecture; Chapter 7, Section 7.2.2 Array Indexed Variables | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 2. | Question : | (TCO 11) Which of the following technique can be used to process the elements of a single-dimensional array? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Selection structure | |  |  | CORRECT | Looping structure | |  |  |  | Sequence structure | |  |  |  | None of the above | |  | Instructor Explanation: | Lecture; Chapter 7, Section 7.2.4 Processing Arrays | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 3. | Question : | (TCO 11) A three-row, four-column array has a total of how many elements? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | 7 | |  |  | CORRECT | 12 | |  |  |  | 16 | |  |  |  | 1 | |  | Instructor Explanation: | Lecture; Chapter 8, Section 8.2 Declaring Two-Dimensional Arrays | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 4. | Question : | (TCO 12) Given the following array:  int profit [5] = {10, 20, 31, 55, 66};   The following statement would replace which value? profit [3] = profit[2 - 1]; | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | 55 with 10 | |  |  |  | 31 with 10 | |  |  |  | 31 with 20 | |  |  | CORRECT | 55 with 20 | |  | Instructor Explanation: | Lecture; Chapter 7, Section 7.2.3 Array Initializers | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 5. | Question : | (TCO 12) In the variable initialization: char course[20] = Welcome to CIS170C ; the character at course[6] will be \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | \0 | |  |  |  | the letter m | |  |  | CORRECT | the letter e | |  |  |  | a space | |  | Instructor Explanation: | Lecture; Chapter 7, Section 7.10 C-strings | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 6. | Question : | (TCO 11) In the following array, what is the value of table[0][0]? int table[4][3]={0,7,3,2,4,9,8,1,3,6,5,4}; | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | 3 | |  |  |  | 2 | |  |  |  | 1 | |  |  | CORRECT | 0 | |  | Instructor Explanation: | Lecture; Chapter 8, Section 8.2 Declaring Arrays | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 7. | Question : | (TCO 11) What type of error will this code produce?   int n[10], i; for(i = 0; i < 10; ++i)      n[i] = i; | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | No compiler errors will be reported. | |  |  |  | A compiler error is generated since there are no braces {} with the for loop. | |  |  |  | A runtime error may occur because we are out of bounds on the array. | |  |  |  | A compiler error is generated since there are no braces {} with the for loop, AND a runtime error may occur because we are out of bounds on the array. | |  | Instructor Explanation: | Lecture and Chapter 7: The correct symbol < was used. You could have gotten an out of bounds had it been the <= symbol. | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 8. | Question : | (TCO 11) Is it possible to write code in this manner for a C++ program?  #define size 20 int main() {      int numbers[size]; } | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | Yes, this is perfectly acceptable C++ code. | |  |  |  | No, the array dimension cannot be a variable. | |  |  |  | Yes, you could write it this way, but you would be wrong. | |  |  |  | None of the above | |  | Instructor Explanation: | Lecture and Chapter 7 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 9. | Question : | (TCO 11) What is wrong with this code?  void WriteDay(char d[]); int main( ) {      char Day [10] = "Friday";      WriteDay(Day[]);      return 0; } | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | The brackets are not needed in the call statement. | |  |  |  | Friday is too short for the array. | |  |  |  | The code is correct as written. | |  |  |  | The call statement should be written as WriteDay(Day[10]); | |  | Instructor Explanation: | Lecture and Chapter 7 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 10. | Question : | (TCO 12) Assuming that t is an array and tPtr is a pointer to that array, which expression refers to the address of element 3 of the array? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | \*( tPtr + 3 ) | |  |  |  | tPtr[ 3 ] | |  |  | CORRECT | &t[ 3 ] | |  |  |  | \*( t + 3 ) | |  | Instructor Explanation: | Lecture and Chapter 8 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 11. | Question : | (TCO 11) Explain how to determine the number of elements in a previously defined single-dimension array. Provide a C++ program segment that illustrates your answer. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | If the array is defined: int array1[] = {0, 5, 3, 6, 8}; C++ automatically knows that the number of elements is: 5, because there are 5 elements in the array (0, 5, 3, 6, 8). If the array wasn't pre-defined, it would have to be coded like: int array1[5]; # include <iostream> using namespace std; int main() { int array1[] = {0, 5, 3, 6, 8}; for (int i = 0; i<5; i++) { cout<<array1[i]<<" "; } cin.ignore(); } This will output: 0 5 3 6 8 | |  | Instructor Explanation: | The number of elements is an array can be determined by using the sizeof function. The sizeof function returns the number of bytes contained in a data item. The number of elements in an array can be determined by calculating the number of bytes the entire array consumes and then dividing that number by the number of bytes an individual element of an array consumes.   Here is a program segment that calculates the number of elements in an array:  int myArray[5] = {1,2,3,4,5}; int arraySize = (sizeof(myArray)) / (sizeof(myArray [0]));   Lecture and Chapter 7 | | |

Week 6

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1. | Question : | (TCO 9) What type of menu first displays the main menu and then, based on the user's selection, displays a submenu? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Single-level menu | |  |  | CORRECT | Multiple-level menu | |  |  |  | Step-level menu | |  |  |  | Sublevel menu | |  | Instructor Explanation: | Lecture | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 2. | Question : | (TCO 9) When the user makes a selection from the main menu in a multiple-level menu, a \_\_\_\_\_ might be displayed next. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | menu | |  |  |  | main menu | |  |  |  | grand menu | |  |  | CORRECT | submenu | |  | Instructor Explanation: | Lecture | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 3. | Question : | (TCO 9) What type of programming encapsulates data and functions together in an object? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | Object-oriented | |  |  |  | Interactive | |  |  |  | Procedural | |  |  |  | Menu-driven | |  | Instructor Explanation: | Lecture; Chapter 9, Section 9.2 Delivering classes for objects | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 4. | Question : | (TCO 9) The variables, arrays, or other data structures that are stored in the object are known as the object's \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | records | |  |  |  | functions | |  |  |  | addresses | |  |  | CORRECT | data fields | |  | Instructor Explanation: | Lecture; Chapter 9, Section 9.2 Delivering classes for objects | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 5. | Question : | (TCO 9) What is the term used for the fields and methods that belong to a class? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Body | |  |  |  | Definition | |  |  |  | Return | |  |  | CORRECT | Members | |  | Instructor Explanation: | Lecture and Section 9.2 defining classes for objects | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 6. | Question : | (TCO 9) You should declare a data field private by using the \_\_\_\_\_ keyword. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | clandestine | |  |  |  | concealed | |  |  | CORRECT | private | |  |  |  | Data fields cannot be private. | |  | Instructor Explanation: | CV 9 9.8 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 7. | Question : | (TCO 9) Which method is called automatically when an object is created? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Accessor method | |  |  |  | Mutator method | |  |  | CORRECT | Constructor method | |  |  |  | Destructor method | |  | Instructor Explanation: | Lecture / Chapter 9, Section 9.3 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 8. | Question : | (TCO 9) Given the following class definition:  class Sample {      public:           Sample() { cout << "constructor called, "; }           ~Sample() { cout << "destructor called, "; }  };  What will the following program output?  int main() {      for ( int i = 1; i <= 2; i++ )           Sample object;      return 0; } | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | constructor called, destructor called, constructor called, destructor called, | |  |  | INCORRECT | constructor called, constructor called, | |  |  |  | constructor called, constructor called, destructor called, destructor called, | |  |  |  | None of the above | |  | Instructor Explanation: | Lecture / Chapter 10 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 9. | Question : | (TCO 9) An Object is destroyed with a \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | destroyer | |  |  | CORRECT | destructor | |  |  |  | deductor | |  |  |  | None of the above | |  | Instructor Explanation: | Week 6 Lecture | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 10. | Question : | (TCO 9) Which of the following is correct? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | A constructor must be declared for each class. | |  |  | INCORRECT | A constructor must be declared with a return type. | |  |  |  | A default constructor can only be provided by the compiler. | |  |  | CORRECT | A class can have only one default constructor. | |  | Instructor Explanation: | Lecture / Chapter 10 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 11. | Question : | (TCO 9) Explain which is more appropriate for writing a menu program: a switch case or if-else statements. Provide a snippet of code to support your position. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | I believe that a switch case like what we used in our iLab6 is the best way to go. It is much easier to read when dealing with menus, and there is far less code. For example: int choice = 0; switch (choice) { case 1: cout<<"You chose choice 1"<<endl; break; case 2: cout<<"You chose choice 2"<<endl; break; case 3: cout<<"You chose choice 3"<<endl; break; default: cout<<"invalid choice"<<endl; break; } vs. int choice = 0; if (choice == 1) cout<<"You chose choice 1"<<endl; else if (choice == 2) cout<<"You chose choice 2"<<endl; else if (choice == 3) cout<<"You chose choice 3"<<endl; else cout<<"Invalid Choice"<<endl; Both options have the same effect, I just think the switch statement is much better for menus in classes. | |  | Instructor Explanation: | Lecture - Menu section    When createing a menu program, both an if-else statement and a switch case could be utilized to procuce functioning code. But, by using a switch case you make your code cleaner, more organized, and easier to read and improve should you need to. You can actually use if-else statements within a switch case statement. When making a menu program, I would use a switch case statement. Here is a sample of a switch case code in a menu program using if-else within them:    switch (input) {      case 1:           cout << "\nCurrent balance is: $" << balance << endl;            break;      case 2:           cout << "\nHow much would you like to withdraw? $";            cin >> withdrawal;           if (balance > withdrawal && withdrawal > 0.0f)            { // is withdrawal amount valid?                balance -= withdrawal; // subtract withdrawal from balance                cout << "\nCurrent balance after withdrawal is: $" << balance << endl;                 // to show balance after withdraw           }            else                 { cout << "\nNot a valid withdrawal"; // notify user of invalid  withdrawal } break; default: cout << "Please enter valid input"; // notify user of invalid input } | | |

Week 7

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1. | Question : | (TCO 13) Sequential files are accessed in \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | random order | |  |  |  | binary order | |  |  | CORRECT | consecutive order | |  |  |  | None of the above | |  | Instructor Explanation: | Lecture / Chapter 13 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 2. | Question : | (TCO 13) In terms of storage and retrieval of data, sequential access files are similar to \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | floppy disks | |  |  |  | books | |  |  |  | hard disks | |  |  | CORRECT | cassette tapes | |  | Instructor Explanation: | Lecture section: What is a sequential file? | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 3. | Question : | (TCO 13) When reading data from a sequential file, the data will be \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | read in alphabetical order | |  |  |  | sorted from smallest to largest | |  |  |  | in key field order | |  |  | CORRECT | in the order in which it was written | |  | Instructor Explanation: | Lecture / Chapter 13 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 4. | Question : | (TCO 13) Text files are what type of file? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | Sequential | |  |  |  | Random access | |  |  |  | Binary | |  |  |  | Consecutive | |  | Instructor Explanation: | Lecture / Chapter 13 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 5. | Question : | (TCO 13) What do the following statements accomplish?  ifstream theFile; theFile.open( myFile.txt , ios::cur); | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Opens myFile in read mode | |  |  |  | Opens myFile in current mode | |  |  | CORRECT | Calculates the offset from current file pointer | |  |  |  | Calculates the offset from the current mode | |  | Instructor Explanation: | Lecture / Chapter 13, Table 13.4 Seek base | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 6. | Question : | (TCO 13) When a file is opened in the append mode, the file pointer is positioned \_\_\_\_\_. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | at the end of the file | |  |  |  | at the beginning of the file | |  |  |  | in the middle of the file | |  |  |  | after the file header | |  | Instructor Explanation: | Lecture / Chapter 13 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 7. | Question : | (TCO 13) Which of the following functions should be used to determine if a file was successfully opened? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | is\_active | |  |  | CORRECT | is\_open | |  |  |  | is\_closed | |  |  |  | is\_ready | |  | Instructor Explanation: | Lecture / Chapter 13 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 8. | Question : | (TCO 13) What is the data hierarchy arranged from smallest to largest? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | INCORRECT | Bit, byte, record, field, file | |  |  |  | Byte, bit, field, record, file | |  |  | CORRECT | Bit, byte, field, record, file | |  |  |  | Byte, bit, field, file, record | |  | Instructor Explanation: | Lecture / Chapter 13 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 9. | Question : | (TCO 13) Given the following declaration:   ifstream theFile;     When using a loop to read lines from a sequential file, what would be the proper loop condition? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | CORRECT | while (!theFile.eof()) | |  |  |  | while (theFile.eof == true) | |  |  |  | while (theFile !eof) | |  |  |  | while (theFile.eof != false) | |  | Instructor Explanation: | Lecture / Chapter 13 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 10. | Question : | (TCO 13) Which of the following is true about files? | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | C++ views each file as a random stream of bytes. | |  |  | CORRECT | Files are opened by creating objects of stream classes. | |  |  |  | Member functions of stream objects cannot be applied to file streams. | |  |  |  | istream, ostream, and iostream are derived from ifstream, ofstream, and fstream, respectively. | |  | Instructor Explanation: | Lecture / Chapter 13 | | |  | | | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 3** | |  | Comments: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Question 11. | Question : | (TCO 13) Provide a C++ code segment that reads data from a text file using a while loop. Use comments to explain what key lines of code are doing. | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | #include <iostream> #include <string> #include <iomanip> #include <fstream>// add to use fstream class using namespace std; // main fuction int main() { char delimiter = ','; // needed for csv file type string item = ""; double price = 0; char ans = 'y'; // initialize object to create text file to write user input to ofstream intoFile("textFile.txt"); do { // get input from user cout << "Enter item: "; getline(cin, item);// place user input into item cout << "Enter price $"; cin >> price;// place user input into price cout<<"Enter another item? (Y or N) "; cin>> ans;// place user input into ans cin.ignore();// get rid of extra return after ans cout<<endl; if (intoFile.is\_open())// check to see if text file is open, if it is, write the user input to the file using the delimiter { // write to the file intoFile << item << delimiter; intoFile << price<<delimiter; } else cout<<"can't find file"<<endl;// this is if the file can't be found (I thought the program makes the file if one wasn't there...) } while(ans == 'y' || ans == 'Y');//if the user typs a y then the code loops to get more user input and appends it into the file. // if the user types an n then the file gets closed and the code continues intoFile.close(); // initialize file to grab info from file to output to program ifstream outOfFile("textFile.txt"); if(outOfFile.is\_open())// if the file is open run the following code { string buffer; int fieldNum = 1; int itemCount = 1; cout << " Sales Receipt " << endl; getline (outOfFile, buffer, ',');// get the first entry in the file and stop at the comma while(!outOfFile.eof())// go through the text file until you reach the end of the file and execute the following code { if(fieldNum==1)// Since there are only 2 parts to each item entry, we check to see if we are in position one or two // if it's position one then execute the following code. { cout << "item # " << itemCount << endl; cout << "Item: " << buffer << endl; fieldNum++;// makes fieldNum equal to two (no longer one), so it moves to the else section below } else { cout << "Price $" << buffer << endl<<endl; fieldNum=1;//puts the fieldNum back to 1 so that we move to the first string before the "," of the next item itemCount++;// updates the count of how many different items there are in the list } getline (outOfFile, buffer, ',');// gets the next string section before the "," in the buffer }// end while outOfFile.close(); // close file after loop ends }// end if else cout << "Error: could not open file" << endl;// in case the file can't be found output error to user system("pause"); return 0; } | |  | Instructor Explanation: | Lecture / Chapter 13  int main () {    string oneLine;    //create a input stream object and open a file    ifstream inMyStream ("c:/myExample.txt");     //check to see if the file opened successfully    if (inMyStream.is\_open()) {        //loop through the file until the end is reached        while (!inMyStream.eof() ){           //read one line from the file           getline (inMyStream, lineBuffer);           //output the line read to the console           cout         }         //close the file stream object         inMyStream.close();   }   else cout return 0; } | | |