**CSI 117 Review for the Midterm Exam**

**Step 1:** Begin by reviewing the objectives listed at the start of each of the PPT lectures. If you do this for all the PPTs (before you begin this review), you will have a good foundation for the Midterm Review.

**Step 2:** Complete the midterm review.

**Step 3**: Study the review you have created and practice problems similar to class and homework exercises.

|  |  |  |
| --- | --- | --- |
|  | **Definitions and Questions *Find the definitions for the following terms in the Gaddis text or the PowerPoint lecture slides.*** | **Answer** |
| **Ex.** | **hardware** | **Hardware consists of the physical devices that make up a computer system.** |
| 1. | software | Programs, commonly referred to as software, all of which are essential to make computers useful. |
| 2. | A computer system consists of 5 types of computer hardware devices.  What are these? | The CPU, Main Memory, Secondary storage devices, Input, and Output Devices |
| 3. | How does a computer store numbers? | Can be represented by the on off position of bits using the binary numbering system |
| 4. | How does a computer store characters? | It is first converted to numeric code, which is then stored in memory as a binary number. The most popular coding scheme being ASCII or American Standard Code for Information Interchange |
| 5. | What is the fetch-decodeexecute cycle? | When a CPU executes the instructions in a program, this the name of that process. The cycle, consisting of these three steps is repeated for each instruction in the program. Feth reads the next instruction fro mmemory into the CPU. Then in the decode the machine language instruction represented is decoded by the CPU to determine which operation it should perform. In the excute cycle that operation is performed, or executed. |
| 6. | syntax error | A syntax error is a mistake such as a misspelled key word, a missing punctiuation character, or the incorrect use of an operator- kind of like a grammar error in the code. |
| 7. | semantic error | A semantic error is a logic error in the program, whereas a syntax error will prevent a program from running, a logic error may exist while thr program runs and lead to unintended results. |
| 8. | compiler | A CPU only understand machine language, therefore high level language languages must be translated after being written by a programmer. For this reason the programmer uses a compiloer to interpret and make the translation. A compiler is the program that translates a high-level language program into a separate machine language program, which can then be executed. |

|  |  |  |
| --- | --- | --- |
| 9. | interpreter | An intepreter both translates and executes the instructions from a high-level programming language. Since combining translation and execution they don’t make a separate machine language program and take longer to execute since they are translating as well. |
| 10. | Describe the 2 steps in designing a program. | Understand the task that the program is to perform.  Determine the steps that must be taken to perform the task. |
| 11. | pseudocode keywords | Pseudocode keywords are terms commonly used to specify actions in a program. |
| 12. | flowchart | A flowchart graphically depicts the steps that take place in a program. |
| 13. | pseudocode | It’s helpful to programmers to write programs in pseudocode before they write it in actual code to avoid mistakes, and iron out details before writing in a high-level language. The word pseudo means fake, so the word’s origin is fake code. It is an informal language that has no syntax rules, and is not meant to be compiled or excuted. So it’s used to creat models, or “mock-ups.” |
| 14. | What 3 steps do computer programs typically follow? | Input is receceived, some process is performed on the input, and then ouput is produced. |
| 15. | variable | Is a storage location in memory that is represented by a name |
| 16. | string literal, character literal, numeric literal | A string literal is a sequence of characters usually enclosed by quotation marks. A numeric litral is a number that is written into a program’s code (Stored as integer or real number). |
| 17. | variable declaration | Because most programming languagers require that you declare all of the variables that you inend to use ina program.. a variable declaration’s purpose is typically to specify two things, a variable’s name and the variable’s data type. |
| 18. | constant | A named constant is a name that represents a value that cannot be changed while the program is executing, and must be declared before they can be used in a program. |
| 19. | module | A group of statements that exists within a program for the purpose of performing a specific task. |
| 20. | local variable | Is declared inside a module and cannot be accessed by statements that are outside the module. |
| 21. | scope | Describes the part of the program in which a variable can be accessed |

|  |  |  |
| --- | --- | --- |
| 22. | module header    After you find the definition, list the elements that must be included in a module header. | Every module must begin with a module header marking the starting point of a module  Most contain the keyword module, the name of the module, and parenthesis ( ) |
| 23. | module call    After you find the definition, list the elements that must be included in a module call. | To execute a module you must call it. When the end of a module call is reached the computer jumps back to the part of the program that called the module. Uses the correct pseudocode keywords, Call and correctly defines the module, Contain no spaces–Contain no punctuation–Must begin with a letter–Use camelCase |
| 24. | parameter    After you find the definition, tell what type of statement may include parameters. | A parameter is a variable that receives the value of an argument passed to it when the module is invoked or called. Parameters are found in a module header and must include a data type and a parameter name. |
| 25. | argument    After you find the definition, tell what type of statement may include arguments. | An argumet is a value that is sent to a called or invoked module, they are found in calling statements. |
| 26. | pass by value | Pass by value means that only a copy of the argument’s value is passed into the module, or one-way communication. |
| 27. | pass by reference | Pass by referenrce means that the argument is passed into a reference variable, meaning it’s two-way communication: the calling module can communicate with called module; and called module can modify the value of the argument. |
| 28. | function | Is a type of module that returns a value back to the part of the program that called it. |
| 29. | library function | They usually perform common tasks and save time for the programmer because it allows for cod ereuse, they are provided by languages and include several libraries of functions such as a Random Number Generator. |
| 30. | function header    After you find the definition, list the elements that must be included in a function header. | Function headers specify the data type of the value that is returned, the elements that must be included in a function header are the name of the function and any parameter variables. |
| 31. | object-oriented software development | A newer method of developing software where programs are centered on creating objects……. |

|  |  |  |
| --- | --- | --- |
| 32. | object-oriented analysis | Or OOA, determines system requirements. |
| 33. | UML | Unified Modeling language |
| 34. | object-oriented design | Or OOD, determines how to meet the system requirements. |
| 35. | class | A class is code that scpefies the fields and methods for a particular type of object. A class in a program creates a template for a user-defined data type that may be used to create objects of that type.  A class cannot contain the main() method. |
| 36. | object    After you find the definition, explain how to instantiate an object. | An object is a software entity that represents something in the real world, such as a customer.  To create a class object (an instance of the class) However, creating an object requires 2 steps, notjust 1 step  Step 1:  Declare a variable to reference the object  Step 2:  Set aside computer memory for the object, using the CellPhone class as its blueprint, and assign the object's reference to the variable |
| 37. | attribute | Features of an object |
| 38. | object’s state | The set of values of all the attributes |
| 39. | method | Modules or functions that process data |
| 40. | Assume that you represent a software company and are preparing a presentation for a potential customer. List and explain the five features of object-oriented software development. Then discuss the advantages of using the objectoriented development process. | A newer method of developing softwarewhere programs are centered on creating objects Object reusability  The five features of object-oriented software development are:  **Encapsulation:** and objects data members or attributes are combined into a single entity  ntire entity can be easily incorporated intomany different programs  **Information Hiding**: details of how a method or class performs its tasks are hidden from the rest of the program  f method or class is changed, rest of programwill not be affected When method is called, programmer worriesonly about interface (inputs and outputs) tomethod, does not know about lower leveldetails  **Data Abstraction**: Process of removing characteristics from something in order to reduce it to a set of essential characteristics.  Programmer hides all but relevant data aboutan object in order to reduce complexity andincrease efficiency Object is a representation of original, withunwanted detail omitted  **Inheritance:** Acquiring the traits of one’s predecessors  Define a base class, or parent class, orsuperclass, then use it to define other classes(child classes or derived classes orsubclasses) Child class inherits properties from the parentclass and may have some additional propertiesof its own  And  **Polymorphism:** ability to create multiple methods with the same name that will act differently when used with different objects  Can also write multiple methods with the samename, but different parameter lists, inside thesame class  A key advantage is object reusability where a object is not a standalone program but can be used by programs that need its service. Saves time- organized. |

|  |  |  |
| --- | --- | --- |
| 41. | visibility modifier | are specified for attributes and methods tocontrol how other parts of the program gainaccess to them |
| 42. | private access | : cannot be accessed by any method that isnot part of the object’s class; data members(attributes) are usually private |
| 43. | public access | can be accessed by other programs andmethods outside the class; methods are usuallypublic |
| 44. | class diagram | Provides a graphicical view of the class. Class diagrams are drawn as an UML diagram to assist in the designing of a class. It is an OO modeling standard. They provide a good overview of a class but not details on the method instructions. |
| 45. | class definition | Set of pseudocode program statements that detail the object’s attributes (data members) and methods. |
| 46. | accessor method | Return information about an object’s current state (i.e., the valueof data members) |
| 47. | mutator method | Change the value of data members |
| 48. | application | Programs that people normally spend most of their time running on their computers performing everyday tasks are referred to as application software.  Programs that make a computer usefulfor every day tasks |
| 49. | describe the 2 steps required to instantiate an object | Step 1:  Declare a variable to reference the object  Step 2:  Set aside computer memory for the object, using theCellPhone class as its blueprint, and assign the object's referenceto the variable |
| 50. | constructor | A constructor is a method that is automatically calledwhen an object is created; i.e., when the New keyword isused to set aside memory to store the object |
| 51. | default constructor | A constructor with no parameters is called the defaultconstructor. |
| 52. | private access | 42 |
| 53. | public access | 43 |
| 54. | List and describe the 4 types of instance methods. | Methods defined in a class are referred to as instance methods.Instance methods are methods for which an object must exist.  ----Accessor methods Return information about an object’s current state (i.e., the valueof data members)  ----Mutator methods Change the value of data members  -----Manager methods Initialize data members (constructors)  -----Helper methods Private methods that support other methods in the classMay do error checking or perform calculationsCalled only by other methods in the same class |

|  |  |  |
| --- | --- | --- |
| 55. | spaghetti code | Spaghetti code: unstructured codethat is confusing and complex dueto poor structure • Flowchart of spaghetti codeshows crossed flow paths with noclear path from beginning to end • Unstructured code is difficult toread, understand, and modifyS |
| 56. | Define the Structure Theorem and describe the three basic control structures used to create structured methods. | Structure Theorem states: It ispossible to write any computerprogram by using only 3 basiccontrol structures:  – sequence one step followed byanother, with nobranching or looping  ,– selection,   choosean action fromamongalternatives basedon a decision,condition, or test  and– repetition  repeat a set ofs teps whilesomething istrue |
| 57. | List the three types of selection structures. | Single-alternative (If-Then)  Dual-alternative (If-Then-Else)  Multiple-alternative (Case) |
| 58. | assignment operator | the equal sign   = |
| 59. | List the relational operators | <, <=, >,>=, ==, != |
| 60. | operator precedence | Is the order of operations for logical operators: such as in arithmetic groupic symbols are evaluated first then exponents, mult, division, add, sub as they appear left to right exactly like the PEMDAS rule. The precedence for logical operators is Grouping Symbols then NOT then AND left to right then OR left to right. |
| 61. | List the boolean operators | NOT, AND, OR |

|  |  |  |  |
| --- | --- | --- | --- |
| **Create a constant name and data type for each of the following.** | | | |
| 66. | minimum wage | MIN\_WAGE | Real |
| 67. | secret 4 character code | PIN | String |
| 68. | sales tax rate | SALES\_TAX\_RATE | Real |

|  |  |  |  |
| --- | --- | --- | --- |
| **Application Problems** | | | |
| **Create a variable name and data type for each of the following.** | | | |
|  | **Description** | **Variable Name** | **Data Type** |
| 62. | number of employees | numEmployees | Integer |
| 63. | middle initial | midInit | Character |
| 64. | telephone number | telNum | String |
| 65. | course grade for a pass/fail course | credGrade | Character |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identify five objects that might belong to the following classes.** | | | |
| 69. | StudentSupplies |  | pencil, notebook, backpack |
| 70. | PackingMaterials |  | box, peanuts, tape |
| **Specify an appropriate class name for each of the following lists of objects.** | | | |
| 71. | colonial, modern, craftsman, bungalow, cottage |  | House |
| 72. | milk, cheese, yogurt, butter, iceCream |  | Dairy |
| 73. | Draw a class diagram by hand for a Pet class that stores the species of the pet, the age of the pet, and the name of the pet. Include the following methods:   1. a default constructor, 2. another constructor method that has a parameter for each data member (this constructor initializes each attribute to the value provided when an object of this type is instantiated), 3. an accessor for each data member, iv. a mutator for each data member, and   v. a method that prints all the information stored about the pet.      Type the class definition (pseudocode) for the Pet service class. ***Note: You will be required to draw class diagrams by hand and type the pseudocode into ANGEL on the Midterm Exam.*** | | |

|  |  |  |
| --- | --- | --- |
| 74. | Start with the class diagram you drew for the previous problem and add an application with a main method that will instantiate some objects of the Pet class.    Type the class definition (pseudocode) for an application. Create a main method that instantiates an object named mrMax of the Pet class using the default constructor. Instantiate a second object named msKitty of the Pet class, providing arguments for each of the parameters in the second constructor.  Include calls to **all** the Pet class accessor and mutator methods for one of the objects. Also, include a call to the method that prints all the information about the object. | |
| **For each of the following 4 expressions, assume that a, b, and c are Integer variables with the values:**  **a = 2 b = 9 and c = 10**  **Specify whether expression will be true or false** | | |
| 75. | a > 5 OR b == 9 | True |
| 76. | b < 9 OR c > 7 AND a > 3 | False |
| 77. | (b < 9 OR c > 7) AND a > 3 | False |
| 78. | a != 4 AND NOT (b <= 9) | False |

|  |  |  |
| --- | --- | --- |
| **Create the positive logic for each of the following expressions.**  **You may assume that w, x, y, and z are Integer variables.**  **Ex1: NOT (x < y)**  **Answer: x >= y**    **Ex2: NOT (z == 9)**  **Answer: z != 9**    **When we use the NOT operator with two or more conditions in parentheses, DeMorgan’s laws are used.**  **These are DeMorgan’s Laws (You may assume that j and k are boolean):**  **One Law: NOT (j AND k) will always be equivalent to NOT j OR NOT k**  **(Notice that two things happen: 1. the NOT operator is applied to both j and k**  **2. the AND operator flips to be an OR operator**    **Ex3: NOT (x < y AND z == 9)**  **Step1—Apply DeMorgan’s Laws NOT (x < y) OR NOT (z == 9)**  **Step2—Apply the NOT operators to create positive logic x >= y OR z != 9**  **Final Answer**    **Another Law: NOT (j OR k) will always be equivalent to NOT j AND NOT k**  **(Notice that two things happen: 1. the NOT operator is applied to both j and k**  **2. the OR operator flips to be an AND operator**    **Ex4: NOT (w >= z OR y < 3)**  **Step1—Apply DeMorgan’s Laws NOT (w >= z) AND NOT (y < 3)**  **Step2—Apply the NOT operators to create positive logic w < z AND y >=3**  **Final Answer** | | |
| 79. | NOT (x > y) | X <= y |
| 80. | NOT (z != w) | Z = W |
| 81. | NOT (x <= 2) | X > 2 |
| 82. | NOT (x > y AND z != w) | Not x > y OR NOT z !=w  X <= y OR z == w |
| 83. | NOT (x == 5 OR y == 5) | NOT x == 5 And NOT y == 5  X != 5 AND y !=5 |

|  |  |  |
| --- | --- | --- |
| **Show the output for each of the following code segments.** | | |
| 84. | Declare String msg = "Term"  If msg == "term" Then  Display msg  End If  Display "end of sequence" | Single alternative  “end of sequence” |
| 85. | Declare String msg = "Term"  If msg == "term" Then  Display msg  Else  Display "String is “, msg  End If  Display "end of sequence" | Dual  String is Term  End of sequence |
| 86. | Declare Integer num = 70  Declare Boolean flag = true  If num >= 70 Then  Display "Hello 1"  If flag == true Then  Display "Hello 2"  Else  Display “Hello 3”  End If  Else  Display "Hello 4"  If flag == false Then  Display "Hello 5"  End If  End If  Display "Done" | Nested  Hello 1  Hello 2  Done |

|  |  |  |
| --- | --- | --- |
| 87. | Declare Integer num = 75  Declare Boolean flag = false  If num >= 70 Then  Display "Hello 1"  If flag == true Then  Display "Hello 2"  Else  Display “Hello 3”  End If  Else  Display "Hello 4"  If flag == false Then  Display "Hello 5"  End If  End If  Display "Done" | Hello 1  Hello 3  Done |
| 88. | Declare Integer num = 69  Declare Boolean flag = true  If num >= 70 Then  Display "Hello 1"  If flag == true Then  Display "Hello 2"  Else  Display “Hello 3”  End If  Else  Display "Hello 4"  If flag == false Then  Display "Hello 5"  End If  End If  Display "Done" | Hello 4  Done |

|  |  |  |
| --- | --- | --- |
| 89. | Declare Integer num = 69  Declare Boolean flag = false  If num >= 70 Then  Display "Hello 1"  If flag == true Then  Display "Hello 2"  Else  Display “Hello 3”  End If  Else  Display "Hello 4"  If flag == false Then  Display "Hello 5"  End If  End If  Display "Done" | Hello 4  Hello 5  Done |
| **Write the appropriate pseudocode for each of the following.** | | |
| **Ex** | **Assume count is an Integer, display the message**  **“Hello” when count is more than 7** | **If count > 7 Then**  **Display “Hello”**  **End If** |
| 90. | Assume itemsPurchased and discount are Real, set the discount to .05 when itemsPurchsed is over 10; otherwise set it to 0 | If itemsPurchased > 10 Then  Set discount = .05  Else  Set discount = 0  End if |
| 91. | Assume j is an Integer, write the boolean  expression that is true if j is within the range from 10 to 50 inclusive | If j >= 10 AND <= 50 Then |
| 92. | Assume j is an Integer, write the boolean expression that is true if j is outside the range  from 10 to 50 inclusive | If j < 10 OR > 50 Then |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 93. | Assume amount is an Integer and discount is Real, use low-end values for range checking to set the discount:     |  |  |  | | --- | --- | --- | | **amount** |  | **discount** | | 0 - 9 | 0 |  | | 10 - 24 | .05 |  | | 25 - 49 | .07 |  | | 50 or more | .09 |  | | If amount >= 50 Then  Set discount = .09  Else If amount >= 25 Then  Set discount = .07  Else If amount >= 10 Then  Set discount = .05  Else  Set discount = 0.0  End If |
| 94. | Assume amount is an Integer and discount is Real, use high-end values for range checking to set the discount:     |  |  |  | | --- | --- | --- | | **amount** |  | **discount** | | 0 - 9 | 0 |  | | 10 - 24 | .05 |  | | 25 - 49 | .07 |  | | 50 or more | .09 |  | | If amount <= 9 Then  Set discount = 0.0  Else If amount <= 24 Then  Set discount= .05  Else If amount <= 49 Then  Set discount = .07  Else  Set discount = .09 |
| 95. | Convert the following pseudocode to equivalent logic using nested If/Then/Else logic. Assume that monthID is an Integer value  Select monthId  Case 1:  Display “January”  Case 2:  Display “February”  Case 3:  Display “March” Default:  Display “Not first quarter of year” End Select | If monthId == 1 Then  Display “January  Else monthId == 2 Then  Display “February”  Else monthId == 3 Then  Display “March”  Else  Display “Not first quartyer of year”  End If |

|  |  |  |
| --- | --- | --- |
|  | **Assume that a class named VIdeo exists and contains attributes for the name of the video and its price. Also assume that appropriate constructors, accessors, and mutators exist for the Video class.** | |
| 96. | Write the pseudocode statement that will instantiate a Video object named thisComedy using the default constructor. | Declare Video thisComedy  Set thisComedy = New Video() |
| 97. | Write the pseudocode statement that will instantiate a Video object named myGame using a constructor to initialize the name to “Bridge” and the price to 12.99 | Declare Video myGame = New Video (“Bridge”, 12.99) |
| 98. | Write the pseudocode statement that will change the price of myGame to 50.00 using the setPrice () mutator method. | Call myGame.setPrice(50.00) |
| 99. | Write the pseudocode statement that will display the price of myGame. | Display “Price is “, myGame.getPrice() |
| 100. | Write the pseudocode statement that will make the name of thisComedy be “Three Stooges” using the setName () mutator method | Call thisComedy.setName(“Three Stooges”) |
| 101. | Write the pseudocode statement that will display the name of thisComedy. | Display “The name is”, thisComedy.getName() |
| 102. | Write the pseudocode statement that will display the name of myGame. | Display “The name of my game is “, myGame.getName()  Display “Name is”,myGame.getName() |
| 103. | **Draw the class diagram by hand and type the pseudocode for this problem.**    An Internet music store sells all of its CDs at $10 each. It rewards customers who buy large quantities of CDs using the following discount table.     |  |  | | --- | --- | | Quantity Purchased | Discount | | 10 - 19 | 20 % | | 20 - 49 | 30 % | | 50 - 99 | 40 % | | 100 or more | 50 % |   This means that a customer who purchases 2 CDs will pay 2 \* 10 or $20.00  A customer who purchases 25 CDs will pay .70 \* (25 \* 10) or $ 175.00 (30% discount)  A customer who purchases 80 CDs will pay .60 \* (80 \* 10) or $ 480.00 (40% discount)      **Draw a class diagram by hand for the following program. Be sure to show the relationship between the components in the diagram and show multiplicity values. Type the pseudocode for the class and the application.**   1. Create a Customer class that contains a customer number, the customer’s name, and the number of CDs purchased. For this class:    1. Include a default constructor that initializes each attribute to some reasonable value. ii. Include another constructor method that has a parameter for each data member. This constructor initializes each attribute to the value provided when an object of this type is instantiated. iii. Include accessor and mutator methods for each attribute.   iv. Include a calculatePurchase() method that calculates and displays the cost of the purchase:     1. Add to the class diagram and write the pseudocode for an application program with a main method that instantiates two objects of the Customer class--one that makes use of the default constructor and one that provides appropriate values for all the parameters in the constructor. For one of the objects:    1. Include a call to display the customer’s number.    2. Include a call to set the number of CDs purchased to 60 iii. Include a call to calculatePurchase() | |