Study Guide #3, CSC254

Fall, 2015

*Chapters 9, 10, 11*

FINAL EXAM WILL BE AT 5:00 PM on Wednesday, December 14, 2016

General Questions

What is the difference between a class and an object? Type of variable and instance of variable

What does instantiate mean?  the creation of an **instance of an object**

Chapter 9

|  |  |  |
| --- | --- | --- |
| Page | Questions | Notes & Comments |
| 329 | 9.2 - 9.4 | 9.2 public class classname(){  9.3, reference variable  9.4 classname objectname = new classname(); |
| 329 | 9.5 - 9.6 | 9.5 Constructors vs. Methods: No return vs. Yes return  9.6 How to get Default; Don’t make a constructor, or build a no arg constructor |
| 333 | 9.7 - 9.13 | 9.7 Operator used to access a data field or invoke a method: . this period  9.8 Call Anonymous object: no name – (new file “input.txt”)  9.9 NullPointerException; invoke a method on a null variable  9.10 array is an object as it has no references. |
|  |  | 9.11 no  9.12 a is undeclared  9.13 false |
| 341 | 9.17 - 9.19 | 9.17 N, N, Y, Y, Y, Y, N(void), Y(static)  9.18 public static void  9.19 No, Yes, method2 is static |
| 347 | 9.20 - 9.22 | 9.20 we have been using the terms "getters" and "setters."  Which of those terms corresponds to "accessor" and which refers to "mutator?" getters = accessor; setter=mutator  9.21 Also, what is encapsulation?  What does the term "private" have to do with it? Encapsulation – a variable only works in the { } it was made in. private makes a variable only work in this. |
| 349 | 9.23 - 9.26 | 9.23 Pass-by-value vs. pass-by-reference (String)  9.24 Swapping values of x and y  9.25 a. you can’t swap primatives B. swapping in arrays. C. Wrapper class d. t1’s i=2 and j=1 \n t2;’s i=2 and j=1; static means shared |
| 353 | 9.27 | 9.27 bcause date[0] is empty, line 4 is null, line 5 is a NullpointException |
| 354 | 9.28 - 9.30 | 9.28 nonimmutable, can still return fields  9.29 immutable, all private types, can’t return pointers  9.30 nonimmutable, has return values |
| 356 | 9.31 | 9.31 I + j is 5  K is 2  J is 0 |
| 358 | 9.32 - 9.34 | 9.32 this – in general refers to current instance. Gets rid of ambiguity. This. NOT used when calling constructor. This. Is calling method and data fields.  9.33 the this in line 6 must come first in the method  9.34 Test.id=45; Test is invalid. |

Chapter 10

|  |  |  |
| --- | --- | --- |
| Page | Questions | Notes & Comments |
| 370 | 10.1 | getLoanDate is non-primative, therefore nonimmutable |
| 373 | 10.2 | Nonimmutable, public variables |
|  |  | 10.7 wrapper class = new Integer (7);  10.8 all but e work  10.9 String s=(Integer.*valueOf*(**"23.4"**)).toString();Integer s=(String. toString (**"23.4"**)). intValue();String s=(Double.*valueOf*(**"23.4"**)).toString();Double s=(String. toString (**"23.4"**)). doubleValue(); |
| 383 | 10.7 - 10.11 | 10.10 3 \n -1  10.11 10 \n 10 \n 16 \n 11 \n 11 \n 17 |
| 384 | 10.12 | Autoboxing – taking out redundant word, autounboxing, redundancy to make a bureaucrat blush. C. is the only one that doesn’t work. |
|  | 10.13 | 3 \n -1 |
| 385 | 10.14 | x is 3  y is 7  z is 10 |

Chapter 11

|  |  |  |
| --- | --- | --- |
| Page | Questions | Notes & Comments |
| What is the difference between "extends" and "implements?"  Inheritance (extends superclass) vs. interface (implements interface) | | |
| 416 | 11.1 - 11.3 | 11.1 True a subclass is a subset of a superclass  11.2 extends  11.3 Interface can create limited multiple inheritance |
| 419 | 11.4 - 11.6 | 11.4 A’s no-arg constructor is invoked; because B extends A. Super is implied in B now.  11.5 – super can go anywhere with a ., otherwise 9) goes first in method  11.6 True, when invoking a constructor in sub, the superclass’s no-arg constrctor is always invoked. |
| 420 | 11.7 - 11.10 | 11.7 False, you cannot override a private method  11.8 False, you cannot override a static method |
|  |  | 11.9 super.superclassname  11.10 super.toString |
| 422 | 11.12 - 11.16 | 11.12 Overloading – methods with same name in same class; overridden – reusing method name in subclass  11.13 Overridden  11.14 Depends where the type is returning to |
|  |  | 11.15 Overloaded  11.16 Avoids mistakes and make it easier for other programmers to read. |
| 425 | 11.17 | What is the relationship between polymorphism and dynamic binding?  Would dynamic binding be necessary if there was no polymorphism?  Polymorphism – a variable of a supertype can refer to a subtype object  Dynamic/Late Binding – (ex: Rectangle l = new Square(5);  Polymorphism is the concept that allows Late Binding to happen |
| 430 | 11.24 and 11.26 | 11.24 You can cast a subclass, but not always a superclass. Casting – changing a variable type. Sub class may have extras. Make an array of superclass to fill with subclasses.  11.26 Y, N, N, N, N, Y, Y, N, N, Y/N, N, N, Y |
| 432 | 11.28 | Java applies these classes by default. |