



>> Let's review for the final. Your final exam is going to be comprehensive. There will be 80 items multiple choice and this will cover all the material presented in modules 1 through 6 and the exam counts 40% of your course grade, so very important endeavor here. Let's talk about a study strategy.

Obviously your lecture slides, example programs and videos will be very important you should go through those slides and example programs. If you need to go back to look at the videos that as well but probably concentrate on the slides and examples there. And then the activities, and quizzes, and projects very fertile ground for review.

The activities were, were a learning exercise, so, so clearly take a look at those. The quizzes, you've had six of those, and so there's 60 items right there you should obviously, be, be very familiar with, with the, content of those, and, and so on. Won't be the same items, but, be, be, be similar probably.

And then, your projects, this was a, a, a sort of a larger learning experience, and, hopefully it brought things together. Your textbook has key concepts, and examples in each chapter. Each of the assigned chapters would be it'd be good for you to review those. The key concepts are, are there in the margins, if you don't understand the key concept there's usually examples right there on that page.

And then finally, I put together a practice exam, its 60 items multiple choice and, it should give you a good idea of, of the type of items that'll be on the final and, and should be very helpful in your preparation. And, we'll go through some of those items, in just a few minutes.

So, as far as the course overview, we had 6 modules an introduction, data and expressions, using classes and objects. And then we learned to write classes our own classes and then we took a look sort of a deep look at, control flow conditionals and loops. And we finished up with some, add additional information on those and, and a few other types of constructs that were conditional in the loop.

So, in Module 1, taking you back now for a bit of review, we, we wrote our first simple Java programs here. You edited, compiled, and, and ran these using jGRASP. We saw how to set a break point and then, run a debug and step through your program. We use Javadoc comments in the program and, ran checkstyle to verify that the the comments and and format were correct and we could generate documentation after that as well.



In Module 2 we looked at data and expressions and, in particular, character strings, escapes sequences. And then this is the first time we looked at variables, and assigning variables of value. In fact, we looked at the eight primitive types, you need to know those. And we also introduced, simple if, if else statements, simple in the fact that they had the simple boolean expressions.

And we looked in depth at the arithmetic expressions and operator precedence, and so on. And this was the first chapter or module where we, actually looked at input from the user this was input from the keyboard. And we also looked at converting data say from an int to a double or a double to an int.

In Module 3, we used some existing library classes, and this is the first time we took up objects and, we had the 8 primitive types from Module 2 and now we are looking at reference types, these are the objects and we created objects. Particularly we looked at the string class, we imported packages and, declare using the import declaration.

We looked at the Random class, the Math class, we looked at, formatting output, primarily using DecimalFormat, and then we also introduced the wrapper classes. Recall int, which is a elementary type if you will as a wrapper class Integer capital I. And then double for instance the primitive, variable, or variable type double, lower case double as a wrapper class Double capital D.

And, it has, all the wrapper classes have some useful, methods in there that you can work with. From there we went on to writing our own classes, up until this point, we'd written a class with a main method. But now we're writing a separate class in a file, and this class actually has, fields, constructors, and methods.

And it's gonna allow us to actually create objects from that class and these objects have state and behavior. We were able to generate UML diagrams for the class. We talked about things such as encapsulation, the anatomy of a method, its parameters, local data, and so on. And we also introduced the idea of a constant, which is declared as public, usually public, but it's, also static and final, could be public or private.

Invoking methods in the same class where we didn't actually have to use the dot modifier. Building classes incrementally we look at that and then also, testing classes in Interactions and using a, driver program. And that's, again, was that class a separate class with main in it, which would essentially exercise the, methods, in the class that we wrote to generate objects.



Then from there, we went on and looked at Conditionals and Loops. This was all in the, vain of con, control flow or flow of control, sequence, selection, and iteration and took a more in depth look at boolean expressions, which, sort of drive a lot of these. And in those expressions, we had operators for equality, relational and logical, recall an equality operator was either not equal or equal, equal.

Relational could be less than or equal, greater than or equal, and so on, and then the logical operators were, where we added two boolean operators or two boolean expressions together or we or them. So add or and we can also not an entire expression. And then for selection we looked at the if statement with optional else and else if and then for iterations we looked at the while statement.

Or while loop as we called it. And this is the, module where we introduced the array list class, and this allowed us to actually store things, multiple items in a single, structure, an array list. We could, it was a list. And it was indexed beginning at zero, so we could get them back and we could also, Actually said items in the list, and so on.

We looked at comparing data. There was some things there. If you're comparing two objects, you usually wanna use a compare to or, the equals method. We looked at how indentation and block statements, were, were used. In fact, the, the target of an if or while is actually a single statement if you want multiple statements in there and need to be enclose in a block, which was an open parenthesis, close parenthesis.

And we've used blocks from the beginning, so that wasn't an issue. But we've brought those details to light as well. That check style requires that of us. Then more details of if and while, and also file input using the file and scanner class. So this will enable us to read in data from a file and, and process it in our program.

Then in our final module, more conditionals and loops. In addition to the if statement, we looked at the switch statement, which was a sort of a more efficient form, although we had a restrictions on, what could be used there. We also looked at the conditional or ternary operator which is a little a small if else, if you will, that could be inside of another statement and so on.



And then there was three additional looping statements, the do-while statement, which had the condition at the end. So you're always gonna go through a do-while at least once. And then the for statement that actually collected all of the things right there in the top if you're gonna go through a sort of a specific number of times.

It's a, a nice concise statement that, that you can use a good bit of the time. The while statement, which we did in the previous module, it's the general loop. These others are slightly more specialized. The most specialized was for-each statement, which was used to go through structures that were iterable and particular the array list.

So we had an array list of strings we can say essentially for each string s and, and the array list we're gonna through it, and it goes through from the beginning to the end. And it always goes from the beginning to the end. So again, it's, a sort of a specific use list, but but very nice.

So we've covered a lot of material. Again I encourage you to look at those, go back and look at all the materials that were provided in the way of slides and, and lecture notes and, and so on, and then all of the assignments you did, you did as well.

So let's take a look at the practice exam. It's, it's provided as two PDF files, the first one has the answers. In this, or rather the first one's without the answers and then there's one with the answers. So you've got one with answers and one without. First, I strongly encourage you to go through the exam without the answers.

And you should probably try, if you think you're pretty prepared after you've gone through all the material, at least once. It would be good to go through this exam just like you're taking the exam. It, it's 60 items, it's a PDF. You can print it out and, you know, circle your answer and so on, it's all multiple choice.

But if you go out without the answers, this will give you an idea of how well prepared you are. And then you should go through, go back and grade yourself, using the exam version with the answers. And as you do that, you need to make sure that you understand why the correct is indeed the correct answer.



That is, if you miss one, and, and a different answer selected there. You need to work through that in your mind so that you understand why the correct answer is, is what it is. And, and the, this will go a long getting you ready for the final exam.

It'll be 80 items instead of 60. But, and they'll be similar in structure and all that sorta thing. They'll be different for the most part, but, this should give you, a really good idea. So, let's look at some of the items on the practice exam, I'm not gonna go through them all, but, we'll just look at, a few of them, to sorta give you a feel for, for what we've got there.

So let's look at the first three here. First one's true or false. It says the semantics of a language define the meaning of a statement in the language. True or false? Well, that's true. Syntax, the form of the language is defined by it's syntax. But the schematics actually give it meaning.

So we think about well formed statements as syntactic correct, but the schematics and language say what the statement really does. Second one there, which of the following is an access modifier? You see the choice is there, hopefully, e jumps at you public, that to access modifiers we use public and private, public says everybody can see it, private says it can only be seen within the class of where the modifiers applied.

So, public is the answer there. And then in number three, which of the following is used to concatenate a string? And what that means is we want to put two strings together. And hopefully, you recall that the plus sign if there's a string on one side or the other, the, the, the one that's not a string is converted to a string and, and they're concatenated together.

So this, this is just a sample of three items, without the answers. And if you bring up the exam with the answers, you will see that they are marked like this. All, all the way through the exam, they're marked like this. And, and, and so on here. So going back, let's go down and look at, I'm not gonna go through all of these.

Here's a couple, let's take a look at 9 and 10. Here we've got x, y, and z are doubles. And we've got this equation here $x = 2y^5 + |z|^2$. And we wanna convert that to code and, lots of choices there. But hopefully, you, you hit on the one, right here looks like b is it?



2 times math dot pow, and this is y raised to the fifth power. + Math.pow, and inside that, Math.abs, we're taking the absolute value of z, and then we're raising it to the second power. Number 10 here is an example of using DecimalFormat. And you see the pattern here, the string pattern for numbers on the left of the decimal point, we want those grouped in threes and separated by commas.

So if it's a small number we wouldn't see a comma, but if it goes over 1,000, up to 1,000 we would 1,000 or more I should say, we would see a comma. And then the on the right side of the decimal point the pattern, we want up to four digits of precision, and that's what the four digits there mean.

And the 000 means that we wanna show always at least three. If there is a fourth digit to the right, or if there's ten digits to the right, we wanna show the first four of them. But in any case, we always wanna see at least three. And so down here, we've got strain x equals, a df.format, and the number we're formatting is 1000.1.

And so this is a decimal format and so we look down here at the choices and hopefully D jumps out at you. We see that we've got the 1,000.1, that's the answer there. But we're showing the two zeroes because is said we always wanna see three digits to the right there.

So, that, that's kinda what we've got. And if we went over here to our key, we can go down here and check this. You see there in d, it was b, that's our answer. And d here is the answer in the number 10. So let's look at one more maybe two more.

Here's one fill in the blank. And the way the blanks are represented, here I've got __13__, this is, this is number 13, this is 14, 15, 16, and 17. And down below is what goes in blank 13. The next one is what goes in blank 13, all multiple choice.

And let's just go through this and I think we can work this without even looking down at the choices. So here we've got class dog, public class dog and it's got two fields. A private double weight, public string name, so two fields there, one's a double, one's a string.



And then the first structure here says public and we've got a blank. Double, it looks like weightin and namein and these get assigned to weight and name. Well this should tip you off that this is a constructor. So there's no type, but the constructor, of course, has the same name as the class So, what goes in 13 would be dog, so down here you would select dog, hopefully.

And then here, we've got a getter, a get weight. And so, this is public something, blank 14 there. Get weight and we're returning weight. Well, what goes in 14? That would be the return type, and we're returning weight. Weight is a double, so we we would wanna put double in for blank 14, and you just select that down here from from the choices.

And for 15 we're setting weight, we're taking in weight in And we say weight = _15_. Hopefully you would know that we wanna send weightIn, that's the parameter coming in to weight. And again you would make that selection down there. There is a getter, _16 says setName to nameIn and here, here we do the set.

And notice we're not returning anything, and so what is the return type on setName? Well, that should be void, since we're, we don't have a return statement, we, we've got re, a void return type, if you will. And so down there in number 16, Hopefully you'd select, we would select void.

And then finally in 17, here we've got public blank two string, and then what we're gonna return here is name plus, weight, plus, weighs, weighs plus weight here. And that's a string and and and what we're looking for the return type here. The return type should be a string, since we're, this is the toString method, it returns a string and again, down here in 17, you would pick that.

So, we go over here, to go up here and find that one. There it is right there you'll see all, all of those selected there, OK? So one more, let's do one more before we play computer. Let's go down here to. All right here, this is a pretty good one here.

This is fairly straightforward, so here we've got a segment of code, and, and it's just a method where we're taking in, x, y and z, they're all ints. And, we're gonna step through this code and obviously figure something out. And below are some questions related to this. And so here we've got and number 36 says myNum is a parameter of the example method.



Well the parameters up here for this method are x, y, and z. So myNum is not a parameter so that would be false. And then here's our first little exercise where we play computer. We're gonna call example with 1, 2, and 3. And we wanna know what is return down here with.

So think of, we're looking up here, x is 1, y is 2, and z is 3. So 1, 2, 3, so we set to 0. If x is greater than 7, x is 1. So is 1 greater than 7? That's false. Is 1 greater than 10? That's false.

Is 1 equal to 3? Call z is 3. That's false. And so, we're just gonna return My Num is zero so down here we would select zero. And now we're gonna call it with 7, 2 and 3. So two and three are Y and Z and seven is the first.

And the seven's the only one that's gonna change in all of these. So seven two and three. And so, we say is 7 greater than 7? That's false. Else if, is 7 greater than 10? That's false. And is 7 equal 3? That's false. So again, we're gonna return 0 there.

In the next one, we've got 20, 2, and 3. So 20, 2, and 3, and so we come in here and we say, is 20 greater than 7? That's true, and so myNum gets assigned y, which was 2. Remember we call this 20, 2 and 3. So so myNum is now two and and we're done with this.

Since this was true, we don't go to the else if. We drop all the way down here. To the the second if here we say is 20 equal to three? Well, that's false so we're going to return two on this one. The only thing that got actually got assigned here was myNum equal y and y was two.

And so that's down here we'll see right here, we would return, we would check 2 there. Last one here is we're calling it with 3, 2, 3. And so now we're saying is 3 greater than 7? That's False, else if, is 3 greater than 10? That's False, and so we go down here Is 3 equal equal 3.

Well, that's true and so in this case we set myNum to 3. And so you go down here, and in this case you would choose three. So, if we zip over here to the key we can go down here and look at these. Let's see, where are they?

Right here. There's that fall, so we return 0, 0, 2, and 3 on these. So, you need to and you need to work through these, these will, this will take you some time, but it'll be well worth your effort as far as preparing for the final exam. So, again, just, just to reiterate, definitely want to, go through the exam first with, without the answers, and then check yourself, and then The most important thing is 3 there makes you understand why the correct answer is what it is.

And, prior to actually going through the practice exam, I encourage you to go back through the notes and all of the other class materials, your quizzes especially. And, and, and get ready, ready for the exam. And then, take this practice exam sort of at the end. This will sort of be your final check to see, see if you're ready.

So, and I've enjoyed the course here. I hope you have, and, good luck on that final exam.