

Assigned Readings from "Java By Abstraction" on a week-by-week basis								
v.1								
KEY: Sec=Section; PT=Programming Tip; JD=Java Details; IMD=In More Depth								
Category	Id	Title	Pg(2ed)	Pg(3ed)	week #	Tues Lect	Thurs Lect	Lab
Sec	1.1	Anatomy of a Program	2	2	1		1	
JD	1.1	Life without Imports	4	4	1		1	
PT	1.1	Pitfalls: Replacing a subpackage name by an asterisk, Assuming that asterisks have an impact on performance	4	4	1		1	
PT	1.2	Style: An alternative layout for braces	5	5	1		1	
PT	1.3	Pitfall: Delaying comments until the code is finalized	6	6	1		1	
JD	1.2	Documentation Comments	7	7	1		1	
PT	1.4	Pitfall: Adding a comment that restates a statement	7	7	1		1	
PT	1.5	Style: Do not add a space	9	9	1		1	
JD	1.3	Statement Categories	10	10	1		1	
PT	1.6	Pitfall: Forgetting to recompile after a modification	11	11	1		1	
IMD	1.1	Languages and Translation	12	12	1		1	
IMD	1.2	The Virtual Machine and Platform Independence	14	14	1		1	
Sec	1.2.1	The Declaration Statement -> Variable Names	16	16	1		1	
PT	1.7	Pitfalls: Assuming that a declared variable is automatically initialized to zero, Declaring all variables at the top	17	17	1		1	
Sec	1.2.3	The Declaration Statement -> Declaration and Memory	18	18	1		1	
IMD	1.4	What Are Bytes Made Of?	19	19	1		1	
Sec	1.2.5	The Declaration Statement -> Primitive and Nonprimitive Types	21	21	1		1	
IMD	1.6	What is a "Type"	24	24	1		1	
IMD	1.7	Assignment versus Equality	26	26	1		1	
Lab	L1.1	A Guided Tour	38	38	1			1
Lab	L1.2	An eCheck Tutorial	38	38	1			1
Lab	L1.3	Using the eCheck Server	38	38	1			1
Lab	L2.1	What is a Development Environment	84	84	1			1
Lab	L2.5	2ed Testing the Environment 3ed L2.3	84	84	1			1
Lab	L2.6	Configuring eCheck 3ed 2.4	84	84	1			1
Lab	L2.10	Integrated Development Environments (ECLIPSE) 3ed 2.8	84	87	1			1
Lab	L2.2	Microsoft Windows 3ed replaced by L2.2	84	84	1			
Lab	L2.3	Linux 3ed replaced by L2.2	86	86	1			
Lab	L2.4	Mac OS X 3ed replaced by L2.2	88	88	1			
Sec	2.1.1	2ed: Computing Paradigms -> Procedural Paradigm 3ed: What is Delegation -> Delegating to a static method	47	48	2	2		
PT	2.1	Pitfall: Using a void method in an expression	49	50	2	2		
IMD	2.1	Delegation within a class	51	51	2	2		
Sec	2.1.2	2ed: Computing Paradigms -> Modular Paradigm 3ed: What is Delegation -> Delegation to an Object	51	52	2	2		
Sec	2.1.3	2ed: Computing Paradigms -> Object Oriented Paradigm 3ed: What is Delegation -> Delegation to an Object	52	53	2	2		
IMD	2.3	Abstraction in General; Delegation by Abstraction	57	57	2	2		
JD	2.1	Paradigms in the Standard Library	58	58	2	2		
Sec	2.2.1	Application Development -> App Architecture	58	58	2	2		
JD	2.2	The Main Class Jargon	59	59	2	2		
Sec	2.2.2	Application Development -> The Client View	60	60	2	2		
JD	2.3	Access Modifiers	62	62	2	2		
IMD	2.2	An Object to a class is like a value to a type	55	56	2		2	
Sec	2.1.4	Case Study: Procuring Bread	56	56	2		2	
PT	2.2	Fallacy: The client is inferior to the implementer	61	61	2		2	
IMD	2.4	"What" and "How" in Hardware Design	62	62	2		2	
IMD	2.5	The Client/Implementer Mix-Up	64	64	2		2	
Sec	2.2.3	Application Development -> Post-Compilation Errors	64	64	2			2
Sec	2.2.5	Application Development -> Ready-Made I/O Components	68	68	2			2
Lab	L2.7	Measuring Elapsed Time 3ed 2.5	89	85	2			2
Lab	L2.8	Approximating Today's Date 3ed 2.6	90	86	2			2
Lab	L2.9	Retrieving System Information 3ed 2.7	84	87	2			2
Sec	2.3.1	Software Engineering->Risk Mitigation by Early Exposure	71	71	3	3		
Sec	2.3.2	Software Engineering->Handling Constants	71	71	3	3		
JD	2.5	Blank Finals	73	73	3	3		
Sec	2.3.3	Software Engineering->Contracts	73	73	3		3	
IMD	2.6	The Case for Private Attributes	77	77	3		3	
Sec	3.1	Anatomy of an API	100	96	4	4		
Sec	6.1	Language Support	218	219	4	4		
PT	6.1	Pitfall: Confusing the empty string with the null string	219	220	4	4		
Sec	6.2	String Handling	223	224	4	4		
Sec	6.4.1	Advanced String Handling -> String Buffer	235	236	4		4	
Sec	6.4.2	Advanced String Handling -> Pattern Matching and Regular Expressions	238	239	4		4	
JD	6.2	The regex package	240	241	4		4	

Category	Id	Title	Pg(2ed)	Pg(3ed)	week #	Tues Lect	Thurs Lect	Lab
JD	3.1	An Obligatory Constructor	101	98	5	5		
PT	3.2	Pitfall: Overload and type casting	107	104	5	5		
Sec	3.2.1	A Development Walk-Through -> The Development Process	107	104	5	5		
Sec	3.2.2	A Development Walk-Through -> The Mortgage Application	108	105	5	5		
Sec	3.2.3	A Development Walk-Through -> Output Formatting	111	108	5		5	
IMD	3.1	Variable Number of Arguments	114	111	5		5	
Sec	3.2.5	A Development Walk-Through -> Input Validation	115	112	5		5	
JD	3.3	Throwing Exceptions Using the Standard Java Library	116	113	5		5	
Lab	L3.1	A Guided Tour of an API	126	123	5			5
Lab	L3.2	A Software Project (Fahrenheit to Celsius converter)	127	124	5			5
Sec	3.2.6	A Development Walk-Through -> Assertions	117	114	6	6		
JD	3.4	Enabling Assertions	119	116	6	6		
PT	3.5	Pitfall: Using asset to validate input	119	116	6	6		
Sec	3.3.1	General Characteristics of Utility Classes->Memory Diagram	119	116	6	6		
Sec	3.3.2	General Characteristics of Utility Classes->Advantages of	119	116	6		6	
IMD	3.2	Input Validity as a Precondition	120	117	6		6	
JD	3.5	Errors and Exceptions	120	117	6		6	
Sec	3.3.3	Case Study: Dialog Input/ Output	123	120	6			6
Sec	4.1	What is an Object?	135	133	7	7		
Sec	4.2.1	The Life of an Object->Birth	138	136	7	7		
JD	4.1	What is "Default" in the Default Constructor?	140	138	7	7		
IMD	4.1	The Copying Model	141	139	7	7		
PT	4.1	Pitfall: Invoking the constructor	141	139	7	7		
IMD	4.2	Skipping Steps 2 and 4	142	140	7	7		
PT	4.2	Fallacy: The object reference can be treated as an integer	143	141	7	7		
Sec	4.2.2	The Life of an Object->Objects at Work	143	141	7	7		
Sec	4.2.3	The Life of an Object->The Object and its Reference	146	144	7	7		
Sec	4.2.4	The Life of an Object->Object Equality	147	145	7		7	
Sec	4.2.5	The Life of an Object->Obligatory Methods	149	147	7		7	
Sec	4.2.6	The Life of an Object->Death	150	148	7		7	
IMD	4.3	Accessors may access values derived from object attributes	157	155	7		7	
Lab	L4.1	Anatomy of an API	162	160	7			7
Lab	L4.2	Anatomy of an API	162	160	7			7
Lab	L4.3	Anatomy of an API	162	160	7			7
Lab	L4.4	Creating Objects	163	161	7			7
Lab	L4.5	What are Stocks?	163	161	7			7
Lab	L4.6	The toString Method	164	162	7			7
Lab	L4.7	The equals Method	165	163	7			7
Lab	L4.8	Static vs Nonstatic Features	166	164	7			7
Lab	L4.9	String Input Source (ie, parsing a string literal using Scanner services)	167	165	7			7
Lab	L4.10	File Input Source (ie, using Scanner services to read a file and read the file's contents)	168	166	7			7
Sec	4.3.1	The Object's State -> Accessors and Mutators	151	149	8	8		
Sec	4.3.2	The Object's State -> Attribute Privacy	155	153	8	8		
Sec	4.3.3	The Object's State -> Objects with Static Features	157	155	8		8	
Sec	4.3.4	The Object's State -> Objects with final Features	159	157	8		8	
Sec	5.1	Selection	174	173	9	9		
PT	5.1	Pitfall: Placing a semicolon after keyword if	175	174	9	9		
PT	5.2	Pitfall: Not using braces after keyword if	176	175	9	9		
JD	5.1	The if-else Operator	178	177	9	9		
PT	5.3	Pitfall: Using == to test the equality of real values	179	178	9	9		
PT	5.4	Pitfall: Using == to test the equality of objects	180	179	9	9		
PT	5.5	Pitfall: Using = to assign a boolean	180	179	9	9		
PT	5.6	Pitfall: Usng mathlike inequalities	181	180	9	9		
IMD	5.1	Nested if statements	186	185	9	9		
JD	5.2	The switch Statement	187	186	9	9		
IMD	5.2	Unconditional Transfer of Control	202	201	9	9		
Sec	5.2	Iteration	188	187	9		9	
PT	5.7	Fallacy: The loop's condition is monitored at all times	190	189	9		9	
PT	5.8	Pitfall: Placing a semicolon after a for	192	191	9		9	
JD	5.3	The while Statement	196	195	10	10		
Sec	5.3.1	Applications -> Exception-Free Input Validation	197	196	10	10		
JD	5.4	The do Statement	198	197	10		10	
Sec	5.3.2	Applications -> File I/O	200	199	10		10	
Lab	L5.3	Revisiting the Notion of Scope (ie, redeclaration of variables within nested blocks)	207	206	10			10
Lab	L5.4	The if Statement (ie, scoping)	208	207	10			10
Lab	L5.5	The for Statement (logic errors, scoping)	209	208	10			10

Category	Id	Title	Pg(2ed)	Pg(3ed)	week #	Tues Lect	Thurs Lect	Lab
IMD	6.1	Compile-Time String Literals	221	222	11	11		
PT	6.2	Fallacy: The concatenation operator coerces both operands	222	223	11	11		
JD	6.1	Automatic Boxing and Unboxing	228	229	11	11		
PT	6.3	Pitfall: Parsing a number with spaces	229	230	11	11		
Sec	6.3	Applications	229	230	11	11		
Sec	1.2.2	The Declaration Statement -> The Integer Types	16	16	11		11	
IMD	1.3	How is the Range Computed?	17	17	11		11	
JD	1.4	Number System	18	18	11		11	
Sec	1.2.4	The Declaration Statement -> Other Data Types PLUS the first para of sec 1-2-5	19	19	11		11	
IMD	1.5	The IEEE-754 Standard	20	20	12	12		
JD	1.5	Booleans Are Not Integers	21	21	12	12		
PT	1.8	Pitfalls: Using double quotes for a char literal, Enclosing more than one character in single quotes	22	22	12	12		
PT	1.9	Pitfall: Using real types when an exact answer is sought	24	24	12	12		
Sec	1.3	The Assignment Statement	25	25	12	12		
PT	1.10	Pitfalls: Omitting the multiplication sign, Using brackets or braces in expressions	28	28	12	12		
IMD	1.8	Expression Trees	30	30	12	12		
PT	1.11	Pitfall: Applying algebraic laws and identities to evaluate expressions	31	31	12	12		
PT	3.1	Fallacy: Only primitive types are passed by value	105	102	12	12		
PT	3.3	Idiosyncrasy: printf precision handling	114	111	12	12		
PT	3.4	Pitfall: Testing equality of real numbers	115	112	12	12		
Lab	L5.2	The Short-Circuit Behaviour (ie, demonstration of the left-to-right evaluation of conjunctions and disjunctions; how the VM will stop expression evaluation at the first opportunity to do so)	207	206	12	12		
Sec	2.2.4	Application Development -> Case Study: The Java Standard Library	66	66	12		12	
PT	2.3	Pitfall: Confusing the JRE with the JDK's JRE	67	67	12		12	
Sec	3.2.4	A Development Walk-Through -> Relational Operators	113	110	12			12
Lab	L5.1	The == Relational Operator	206	205	12			12
Lab	L6.1	The Masquerade (behaviour of String vs ints)	243	247	12			12
Lab	L6.3	Symmetric-Key Cryptography	245	249	12			12
Lab	L6.2	Exploring the String Class	244	249	12			12
JD	2.4	Handling Console I/O in Earlier Releases of Java	71	71	x	x		
Sec	2.3.4	Case Study: Meet the Managers	77	77	x	x		
Lab	L2.11	Using UniCon 3ed L2.9	84	89	x	x		
JD	3.2	Static Imports	103	100	x	x		
Sec	6.4.3	Command-Line Arguments		241	x			