Use the following outline to guide your self-assessment and notetaking

Week 3 – Introduction to Methods, Classes and Recursion

Classes and Objects (Ch 4.1)

Scope of a variable

- Depends on the location of the variable
- Ex: A variable declared outside of any method is available to the whole class, a variable created inside a method is only available inside that method and a variable created inside a structure such as an if statement or loop is only available inside that structure.
- It is possible to use a local variable with the same name as a variable declared at class level because these variables are separate by scope. However, do not do this, as it can cause confusion to anyone reading the code.

Anatomy of a Class (Ch 4.2)

Instance Data

- New memory space is reserved for a variable whenever an instance of a class is created
- Attributes make up the object and hold data describing it. This is often referred to as the object's state

UML class diagrams

- Used to visualize relationships between classes and objects
- Each object has attributes and methods indicated in the diagram

Encapsulation (Ch 4.3)

Visibility modifiers

- defines the scope in which a construct can be accessed.
    - The Java visibility modifiers are public, protected, private, and default (no modifier used)
- Public means that the member of a class can be referenced outside of the class (violate encapsulation)
- Private means it can only be used within a class definition (enforce encapsulation)
- Protected is used when classes are inherited

Accessors and Mutators (Getters and Setters)

- Used to manage the data in a class in a controlled way
- Setters are methods that allow a user to modify a value within a class
- Getters return a value in a class

Anatomy of a Method (Ch 4.4)

return statement

- Used to return a data value in a method

- Can be a primitive type, object, or nothing at all (void)
- Methods all have a return type that is specified in the header and the expression must be consistent.

Parameters and Arguments

- Value that is passed into a method when invoked
- Specified after the parentheses in the method header (formal parameters)
- Consist of a name and a data type
- Values passed into a method are the actual parameters (arguments)

Local Data

- A variable can be declared inside a method which makes it local (cannot be used outside of it)
- Formal parameter names are local data
- Any reference to local data outside of the method or class will cause a compile-time error

Constructors Revisited (Ch 4.5)

- Used to initialize (instantiate) an object
- Takes in parameters to instantiate the object
- Does not have a return type (but not void either)
- Each class has a default constructor that does not take any parameters

Boolean Expressions (Ch 5.1)

Equality and Relational Operators

Operator	**Meaning**
==	Equal to
!=	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to

These have lower precedence than arithmetic operators

Logical Operators

A	B	A && b	!a	A || B
true	true	true	false	true
true	false	false	false	true
false	true	false	true	true
false	false	false	true	false

The if Statement (Ch 5.2)

If

- Used to execute code whenever a condition is true
- Processing continues with the next statement
- If condition is false, processing goes to the next statement immediately

if-else

- These are a chain of statements that execute depending on condition
- If condition is true, then the if block is excuted. Otherwise, the else statement is executed

Using block statements

- A collection of statements that do a different thing depending on the condition.
- Enclosed in braces

Nested if statements

- If statements can be enclosed inside of other if statements
- Used to make another decision after determining the result of a previous decision

Comparing Data (Ch 5.3)

Comparing floats (this includes doubles)

- 2 floating point values are equal only if the binary digits of their representation match
- However, in computation, they may not fully match, which is why it is better to use a tolerance level to compare them
    - Ex: Math.abs(f1-f2) < 0.00001 will be true as long as the first 4 values after the decimal are the same

Comparing characters

- Characters are compared lexicographically based on the Unicode character set. (A < B, 0<4, etc)
- Char is a primitive value that represents 1 character
- If you want to compare strings, the comparison is governed by rules for comparing objects

Comparing objects

- You should not use equality or relational operators to compare string objects
    - These check if the 2 strings refer to the same object, not if the characters themselves are the same
    - There are specific methods like compareTo or Equals to check if the strings are the same or not

Recursive Thinking (Ch 12.1)

Infinite Recursion

- This is when the base case never gets executed in a recursive algorithm, causing an infinite loop
- A recursive algorithm must have a base case to stop the loop

Recursion in Math

- An example is the factorial n * (n-1) … 3*2*1
    - Here 1 is the base case

Recursive Programming (Ch 12.2)

Tracing recursion

- Every recursive call to a method creates local variables and parameters
- This process is continued until the base case is reached in the method
- Whenever a method is called, a new data space is also created

Recursion vs Iteration

- Iteration (with loops) tends to be easier to implement then recursion, but it may not be intuitive in all cases
- Iteration is faster because there is no overhead of calling the same method repeatedly

Direct vs Indirect

- Direct recursion is when a method calls itself
- Indirect recursion is when a method invokes another method, resulting in the original method being invoked again