Coding 6.5

IDENTIFIER

Identifier

Just like every entity in the real world has a name, so you need to choose names for the things you will refer to in your program.

Rules for naming identifiers:

- An identifier is a sequence of characters that consists of letters, digits, underscores '_' and dollar sign $.
- An identifier must start with a letter, an underscore or a dollar sign. It cannot start with a digit.
- An identifier cannot be a reserved word.
- An identifier cannot be true, false or null.
- An identifier can be of any length.

Example of legal identifier:
$2, ComputerArea, area, radius

VARIABLES

Variables are used to store data in a program. Variables are for representing data of a certain type. To use a variable, you declare it by telling the compiler its name as well as what type of data it represents. This variable declaration tells the compiler to allocate appropriate memory space for the variable based on its data type.

The syntax for declaring a variable is:
datatype variableName;

Example:

```
public class ComputeArea {
public static void main(String[] args) {
double radius;
double
double area;
radius = 20;
area = radius x radius x 3.14;
System.out.println (area);
}
}
```

In our ComputeArea program (as above), we declare radius and area to be double variables.
Other data types are: int, double, char, byte, short, long, float and boolean.

CONSTANTS

The value of a variable may change during the execution of the program, but a constant represents permanent data that never changes.

In our ComputeArea program, we can declare 3.14 as a constant value and we can name it as PI because 3.14 represent Pi value.

Syntax for declaring a constant:

final datatype CONSTANTNAME = VALUE;

Example:

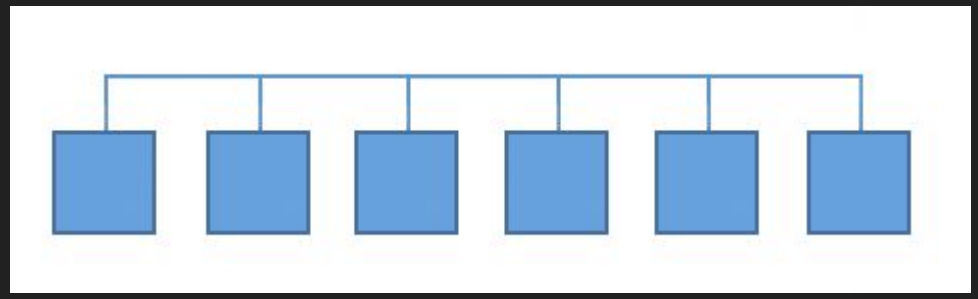
```
public class ComputeArea {
public static void main(String[] args) {
    final double PI = 3.14;
double radius;
double area;
double area;
radius = 20;
area = radius x radius x PI;
System.out.println (area);
```

Taxonomy

taxonomy is a system for naming (labeling) and organizing things into groups that share similar characteristics.

FLAT TAXONOMY

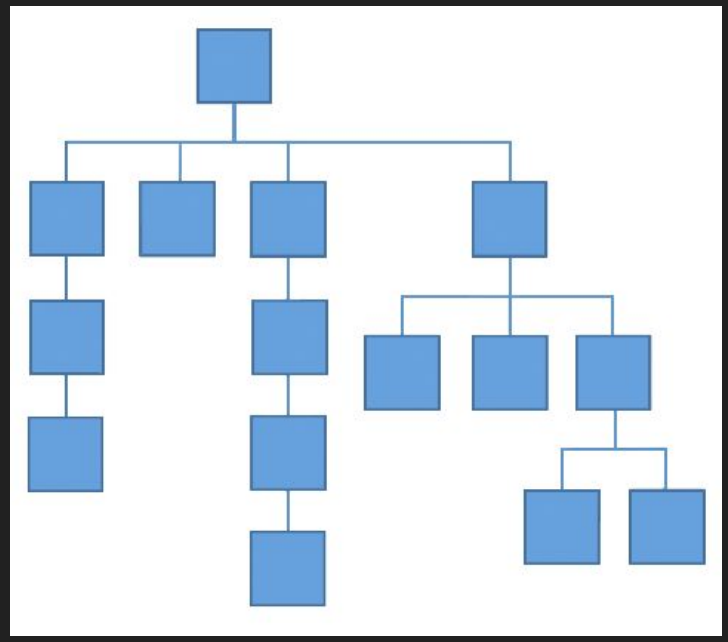
A flat taxonomy, also known as an unlayered taxonomy, is simply a list of items. A flat taxonomy has only top-level categories. In a flat taxonomy, the items are weighted equally, though on a website, it is common to put the most important item first on the list.



HIERARCHICAL TAXONOMY

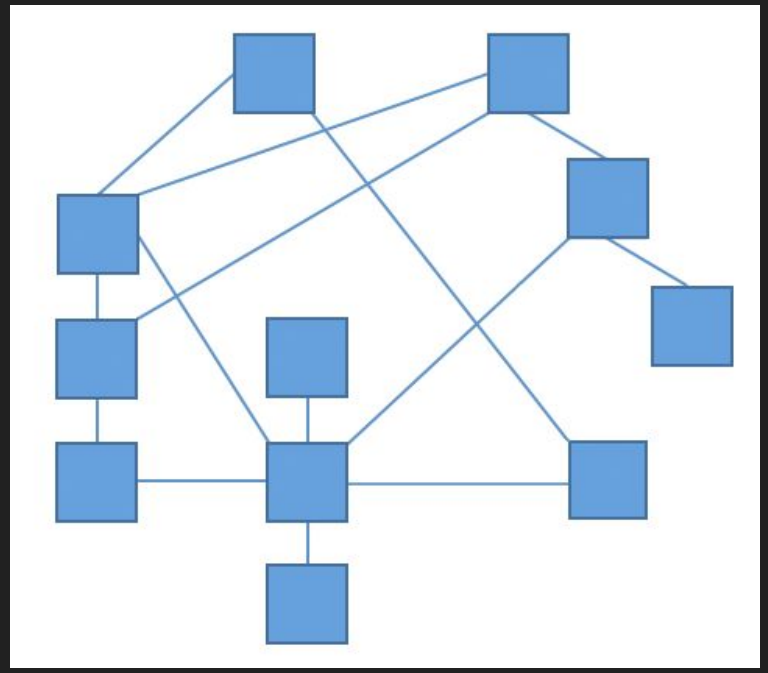
Hierarchical taxonomy is a hierarchical arrangement of categories within the interface of a website or intranet. It is often represented as a tree or a flowchart.

Individual items within the hierarchy are arranged in order of importance or status. Moving up the hierarchy means expanding the category or concept. Moving down the hierarchy means refining the category or concept.



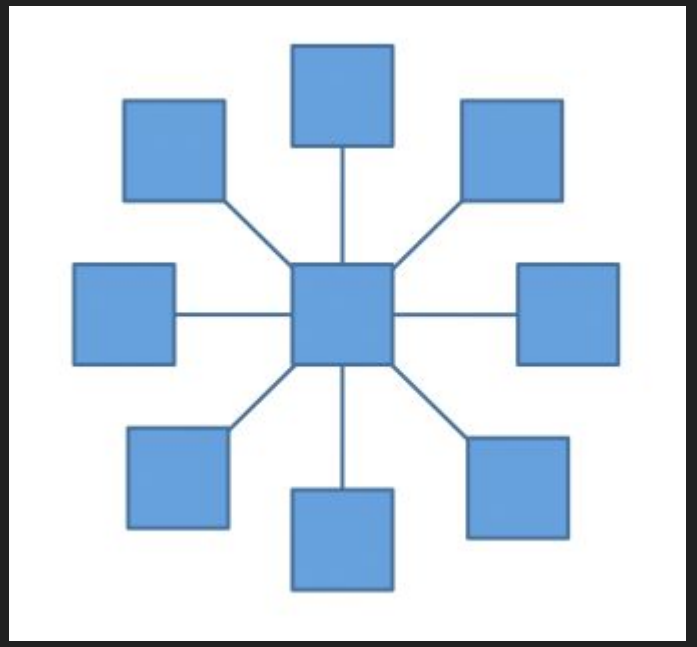
NETWORK TAXONOMY

A **network taxonomy** organizes content into both hierarchical and associative categories. Categories can be linked to any other categories. And relationships among items can have different meanings, including semantic ones.



FACET TAXONOMY

A **facet taxonomy** allows an item to be assigned to multiple taxonomies (sets of attributes), enabling the classification to be ordered in multiple ways, rather than in a single, predetermined order (as in a strict hierarchy).



Useful Links

- https://marketingland.com/website-taxonomy-guidelines-tips-127706

Methods of creating websites

WYSIWYG

Editor or program is one that allows a developer to see what the end result will look like while the interface or document is being created. Example: Microsoft Front Page; Adobe DreamWeaver

WYSIWYG

Pros	Cons
They are easy to use, so even people who have no knowledge of HTML can use them to create their websites.	Most WYSIWYG include HTML code that is hard to read that usually doesn't comply with Web coding standards set forth by the World Wide Web Consortium, also known as the W3C.
Gives you more creative control as you get to focus more on what the design looks like instead of what the HTML code looks like.	Usually the codes are very specific to the product you are using to generate them, which may create issues with their viewing, or editing.
You will save time because a lot of things that take long time to hand code in HTML are quickly and easily done with a few clicks of a mouse.	It will be harder to market yourself to employers who want you to know HTML, not a specific WYSIWIG editor. You won't be able do well if you are used to one particular program and are forced to go to using organic HTML with notepad.

Text based

Editors that require the developer to enter descriptive codes and do not permit an immediate way to see results of the markup. Example: Notepad

Text based

Pros	Cons
Faster to Edit For simple edits, it is often faster to make changes to a page using a text editor.	Must Know HTMLWhile most HTML text editors can help with tags and suggest attributes and so on, this is no substitute for knowing HTML. Most modern text editors have drag and drop styles such as bold and italic, but if you can't remember the code for "non-breaking space" your editor might not be able to help.
Helps You Learn HTML Text editors teach you to read HTML. They often have wizards and functions to do the more common tasks (like the basic page tags), but you'll learn HTML if you use a text editor	Steeper Learning CurveBecause you have to learn both HTML and the editor functions itself, a beginner will find a text editor more difficult to use.
More MarketableA Web Developer who can write HTML using a text editor will be more marketable than one who can only use a WYSIWYG editor. The former is more flexible and can get up to speed on any HTML editing tool, while the latter has to start all over with each new editing tool.	Harder to "Design" WithSome people find text editors more difficult to design pages in because they can't visualize how the page will look from just the HTML.

CMS (Content Management System)

Is a software application or set of related programs that are used to create and manage digital content. Example: WordPress, Sharepoint, Joomla and etc.

CMS

Pros	Cons
Content editing is kept separate from design and functionality of the site; and a typical CMS allows non-technically trained users to add, format, and edit content on a website, without disrupting its design and coding.	Unfortunately, there are some malevolent hackers out there who can figure out how to break into these platforms; so security will require extra precautions.
Each user can be assigned selective access permissions based on their roles (for example, you may choose to allow some users to only add and edit their own content, while giving others universal access).	Making your website look exactly how you want can be more of a challenge. This is true of some CMS frameworks more than others, but all present a bit more work to 'style'.
Content can be updated rapidly; turnaround time for your site updates is generally much faster using a CMS.	The CMS stores everything separately, then assembles it on the fly when the web client requests a page, which means they can be slow; however, this can be mitigated by using strong, effective caching and Content Network Distribution (CDN) systems.

Indentation

- https://codehs.gitbooks.io/introcs/content/Programming-with-Karel/how-to-indent-your-code.html