# DATA ENCAPSULATION

***John Kimutai***

***11.30-1.30***

* Encapsulation is one of the fundamentals of Object-Oriented Programming
* Refers to the bundling of data with methods that operate on that data.
* Used to hide the values or state of a structured data object inside a class, preventing unauthorized parties’ direct access to them.
* Encapsulation is a good idea for several reasons:
* The functionality is defined in one place and not in multiple places
* It is defined in a logical place- the place where the data is kept.
* Data inside an object is not modified unexpectedly by external

code in a completely different part of the program.

* When using a method, one only needs to know the end result of

the procedure.

# *Key terms:*

**Abstraction:** A technique for arranging complexity of computer systems so that functionality may be separated from specific implementation details.

**Accessor:** A method used to return the value of a private member variable.

**Encapsulation:** A language mechanism for restricting direct access to some of an object’s components.

**Information hiding:** The principal of segregation of the design decisions in a computer program from other parts of the program. (encapsulation)

**Mutator:** A method used to control changes to a private member variable, also known as setter method.

**Private:** An access modifier that restricts visibility of a property or method to the class in which it is defined.

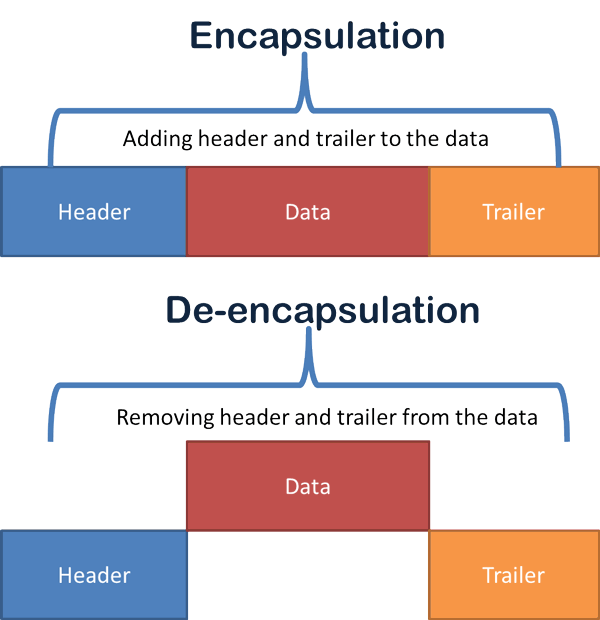
**Public:** An access modifier that opens visibility of a property or method to all other classes.

# DATA DE-ENCAPSULATION

Refers to the process in which information added through the encapsulation process is removed.

Protocol information can be added before or after data. If it is added before it is called a **header.** If it is added after, it is called a **trailer**.

The following diagram represents the process of data encapsulation and de-capsulation process:



Header and trailer added by the sending computer can only be removed by the peer layer of the receiving computer.

Data encapsulated by a layer on the sending computer is de-capsulated by the same layer on the receiving computer through a process known as *same layer interaction.*

This is done through the OSI & TCP/IP models.

After doing encapsulation each layer uses a specific name or term to represent the encapsulated data

The table below lists the terms used by the layers in both models to represent the encapsulated data

|  |  |  |
| --- | --- | --- |
| **TERM** | **OSI layer** | **TCP/IP layer** |
| Data | Application | Application |
| Data | Presentation |
| Data | Session |
| Segment | Transport | Transport |
| Packet | Network | Network |
| Frame | Data link | Data link |
| Bits | Physical | Physical |

Some functions of data encapsulation include:

* Proper data sequencing
* Error detection and control
* Flow control
* Congestion control
* Routing information