

ICT167 ANS5

Constructors

- Special method designed for initializing instance variables in a class
- Every time an object is created (by a client) using new, the constructor is called
- We need to create multiple constructors to allow clients different ways to initiates/create object class

- No return a value or even void
- Constructor Name = Class name
- Must have parameter
 - String str = new String("Information technology")

Default constructors:

- A special type of constructor where there is no parameters and does not refer to any instance variables
 - String str = new String()
- Assigns instance variables default values

Overloading:

- Using the same name for two or more methods in the same class for closely related methods
- Allows the same name methods to take in and return different parameter types
 - Math.max(2,3)
 - Math.max(2.5,6.5)

- No instantiating the overload class/class with overloaded methods if we declare overload method as static
- Parameter (or called method's signature) must be different/different type (eg: double, Integer...)

Process of overloading and automatic type conversion:

- Java tries to find the exact type match between the method signature (think the variables when we invoke a method) and the arguments (method defined in the class and the parameters)
- When an exact type match can't be found the method signature will be converted to match the argument. Rules → int → double not the other way around

Dangers of returning objects (privacy leaks- Class Type Instance Variables):

- The problem arises because the variables will store the memory address of where the object is stored in memory which causes issues such as changing unintended variables in class and access hidden variables in the object.

Solution of privacy leaks:

- Use wrappers to convert class type variables to primitive type variables to return. So this ensures that only primitive values or strings are returned in a method
- Utilise cloning so you make a copy of object and use that

Enumerations:

- Enumeration is (class) data type that controls the possible values that an attribute may contain. So limits values that a variable may hold
 - Enum movieRating{EXCELLENT, AVERAGE, BAD} NOT 1 star out of 5 etc
 - [Object variable].EXCELLENT
- The object referenced will still have the methods available with it

Refer to 13:40 for how to create client class