

AP Computer Science A Study Guide Unit 2

From Simple Studies: <https://simplestudies.edublogs.org> &
@simplestudiesinc on Instagram

Using Objects

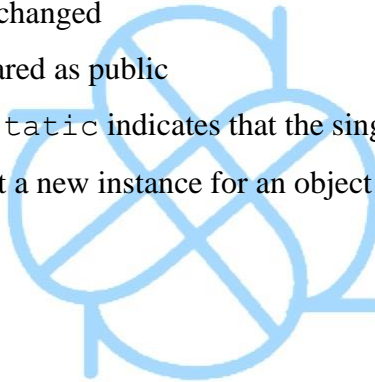
Objects: Instances of Classes

- Object – characterized by state, attributes, and behavior.
 - Instance of a class
- All OOP (Object-Oriented Programming) languages try to represent an object as a variable or an instance in a program.
- Class – Software blueprint
 - Implement Objects of a specific data type
- Data fields –
 - Also known as instance variables
 - Methods
 - Provide behaviors of the object
 - Provide operations which manipulate the object
- Encapsulation – Combining data and method into a single unit
- Public
 - Class declaration which suggests that the class can be used by all client programs
 - Public methods – any client program can access these methods

Objects: Instances of Classes (continued)

- Private
 - Information hiding in Java – restricting access to data and methods

- Private methods and variables – Only accessible by methods from that class
- Java does allow public instance variables but we only have private instance variables in the AP Java Subset.
- Static
 - Static variable – class variable
 - Shared through all instances of a class
 - Static – the memory allocation can only occur once
 - Uses of a static variable
 - Accumulation of a total
 - Providing a new identity for each instance or object in a class
 - Keeping track of data of the objects in a class
 - Static final variable (also known as constant)
 - Cannot be changed
 - Often declared as public
 - Keyword `static` indicates that the single value applies to the entire class and not just a new instance for an object in the class.



Methods

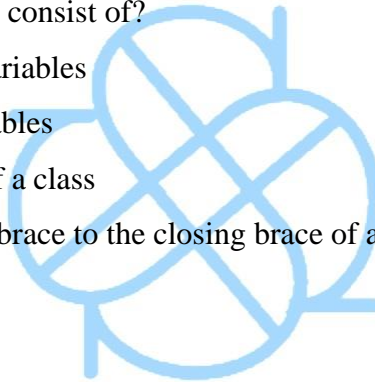
- Method headers look like this:
 - Except constructors and static methods
 - `public void deposit (String password, double amount)`
 - `public` – access specifier

- void – return type
- withdraw – method name
- (Strong password, double amount) - list of parameters
- Types of methods:
 - Instance methods
 - Constructors, accessors, and mutators
 - Constructors
 - Creates an object or a new instance of the class
 - Default constructor – No arguments
 - Provides initial values for each new object
 - Constructor with parameters have instance variables set to the values of the parameters.
 - Object variables store the values and addresses of their respective objects.
 - Accessors
 - Access a class object without modifying an object
 - Returns information about the object
 - . (dot) operator signals a method of a class
 - Mutators
 - Alters the state of an object
 - Changes at least one of the instance variables of the object
 - Invoked similar to an accessor in a client program

Methods (continued)

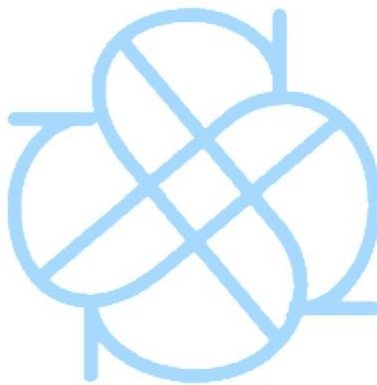
- Static Methods
 - Performs an operation for the whole class
 - Not individual objects
 - Sometimes known as a class method
- Static methods vs. Instance methods
 - Static methods use the keyword static for its implementation.
 - No implied objects in static method (there is in instance methods)

- A static method can use a static variable in the code.
- Method Overloading
 - Overloaded methods are two or more methods which are in the same class and have the same name but different inputs or parameter lists
 - Compiler decides which method to call by looking at the method's signature
 - Method signature – consists of method's name, parameter types
 - Return type is irrelevant for overloaded methods
 - Two methods can have similar method signatures but not different return types (compiler will give you an error)
- Scope
 - Scope of a variable or method means that it's in an area where the identifier is visible and can be easily accessed
 - What does a scope consist of?
 - Instance variables
 - Static variables
 - Methods of a class
 - From the opening brace to the closing brace of a class.



Methods (continued)

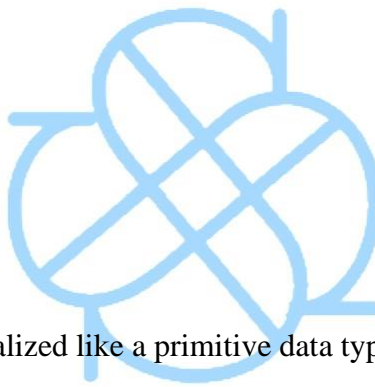
- Local method
 - Inside a method
 - Can be defined in a statement
 - Scope extends from the beginning to the end of the block it is defined in
 - Block – code enclosed in { }
 - The memory of a local variable or method is recycled when a block is exited
 - Higher precedence over instance variables
 - Using same names creates ambiguity for programmers and coders (Leads to more errors)



String Objects: Concatenation, Literals, and More

- String – an object of this type is just a sequence of characters.
- All string literals are implemented as instances of a class.
 - String literals have zero or more characters in them.
 - Can also include escape sequences
 - String literals are surrounded by double quotes.
 - Quotes are not part of the string object.
- String Objects – immutable
 - No methods can change them after they are constructed
- You can create a new String object which can be a modified form of an existing String object.
- String objects can be initialized like a primitive data type.

- Concatenation operator (+)
 - Used on String objects
 - Allows you to combine two or more String operands
- Characters in the String objects
 - Compared based on their position in the ASCII chart.
- Don't use (==) for testing or comparing String Objects.



String Methods

- String objects can be initialized like a primitive data type.
- Two ways to compare String objects:
 - Using the equals method
 - If (string1.equals(string2))...
 - Using the compareTo method
 - Int compareTo (string otherStringYouAreComparing)
 - stringOne. compareTo (stringTwo) < 0
 - stringOne precedes stringTwo in a dictionary
 - stringOne. compareTo (stringTwo) > 0
 - stringTwo precedes stringOne in a dictionary
 - stringOne. compareTo (stringTwo) == 0
 - both Strings are identical
 - REMEMBER THAT JAVA IS CASE-SENSITIVE!

Wrapper Classes: Integer and Double

- Integer Class
 - This wraps a value of `int` type in any object
 - Only contains one instance variable with the type `int`
 - Int methods you should know for the exam
 - `Integer (int value)`
 - To construct an integer
 - `int compareTo (Integer other)`
 - `int intValue()`
 - `boolean equals (Object obj)`
 - Overrides equals in class `Object`.
- Double Class
 - This wraps a value of `double` type in any object
 - `double (double value)`
 - `double doubleValue()`
 - `int compareTo (Double other)`

- o `boolean equals (Object obj)`
 - o `String toString()`
- Both Integer and Double objects contain no mutator methods in their classes.

Using the Math Class

- Methods you need to know in the Math Class
 - o `static int abs(int x)`
 - o `static double abs(double x)`
 - o `static double pow(double base, double exp)`
 - o `static double sqrt(double x)`
 - o `static double random()`