AP Computer Science A Study Guide Unit 5

From Simple Studies: https://simplestudies.edublogs.org &

@simplestudiesinc on Instagram

Writing Classes

Anatomy of a Class

- 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 – Defines another abstract data type in the program
- Object references
    - String variables
- Instance Variables
    - Attributes and behaviors
    - Hold data for objects
- Methods – Code for behaviors or any actions that apply to the objects.
- Constructors – Used to initialize the instance variables
    - When an object is created
- Main Methods – Used to test the class
- Instance variables
    - Attributes/behaviors
    - Fields
    - Properties

Anatomy of a Class (continued)

- Types of methods: Accessor, constructors, and mutators

- Void return type – Indicates that the method doesn’t return any value or String object.
- Method names are always followed by parentheses.
  - Possible parameters should be indicated here.
- Example of a class:

```
public class Name {

   private String first;
   private String last;
   public Name(String theFirst, String theLast) {
      first = theFirst;
      last = theLast;
    }
}
```

- Keywords public and private are used for:
  - Classes
  - Data
  - Constructors
  - Methods

Constructors

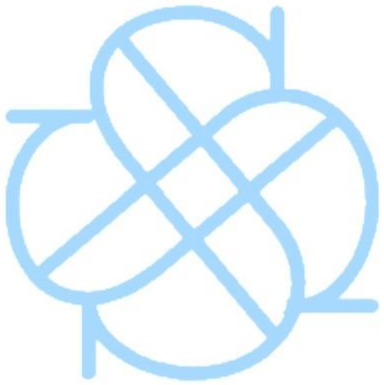
- Constructors – used to set the initial value for an object
  - Or instance variables
- When there is no constructor coded, Java includes a default constructor
- Default constructors don’t have any arguments.

    - Instance variables are set to a default value.
        - Int and double – 0
        - Strings – null
- Constructor parameters are known as local variables to the constructor.
    - They also provide data to initialize instance variables.
- Classes often have multiple constructors.
    - A constructor which has no parameters
    - A constructor with parameters needed for initializing any instance variables.

Documentation with Comments

- // - single line comment
- /* */ - multiple line comment
- /**   */ - documentation comment
- These comments help you remember any changes or additions to the program.
    - It is a good habit to develop for programmers.
- You can use the Java tool known as Javadoc for this.

- Comments are ignored by the Java compiler.
- Preconditions
    - Condition must be true before the code is implemented.
- Postconditions
    - Should be true after a method is run
    - Describes the output/outcome after the method is run
        - Can show any changes occurred to the instance variables
- These conditions help determine the validity of the program/software.
- Software designers and programmers usually use this:
    - Use-case diagram system
    - Shows different ways a user can interact or use the program before its built



Documentation with Comments (continued)

- Example with both types of conditions

```
/**
  * Constructor that takes the x and y position for the snake
  * Preconditions: parameters x and y are coordinates from 0 to
  *    the width and height of the world.
  * Postconditions: the snake is placed in (x,y) coordinates
  * @param x the x position to place the snake
  * @param y the y position to place the snake
```

```
 */
public Snake(int x, int y)
{
  xPos = x;
  yPos = y;
}
```

Accessor Methods

- Also known as get methods or getters
    - How to get the value of an instance variable
- Return by value
    - Original value can’t be modified
    - A way to access the instance variables of the class
- Non-void method returns only a single value.
    - The header has the return type instead of the keyword void
- Accessor methods return primitive types
- Return keyword/ expression
    - References an object

    - Returns a copy of the reference
        - Not the original object
- toString method
    - Overridden method
    - Incorporated in classes to show a description of the object
    - Called when print statements are passed as objects

Mutator Methods

- Also known as a set methods or setters
    - Allows changes to the values of instance variables
- Void methods don’t return a value but they do take parameters for instance variables

Writing Methods

- Procedural Abstraction: Can name a method and call it whenever it’s needed
  - Creating methods
- Programmers break down larger programs into a smaller one.
- To write methods, you need a
  - Method definition
  - Method Signature
  - Method body
- Object.method()
  - To call an object’s method
- Actual parameters can be a primitive value or a reference to the object.
- Why should you use multiple methods in your code?
  - Organization and reducing complexity of the code
  - Reusing code
  - Maintainability and debugging

Static Variables and Methods

- Static variables and methods use the keyword static.
  - Must be before header or declaration
  - Can be public or private
- Static variables
  - Belong to class
  - Objects share single static variables
  - Used with class name and dot operator
- Static methods
  - Associated with class
  - Cannot access or modify any values of the instance variables
  - Can modify the values of static variables.
  - Can’t call non-static methods

Scope and Access

- Scope of a variable – where a variable can be accessed or used
    - Determined by the declaration of the variable
- Java has 3 distinct levels of scope which are related to different types of variables
    - Class level scope
        - Instance variables
    - Method Level Scope
        - Local variables
        - Parameter variables
    - Block level scope
        - Loop variables
- Formal parameters or variables should only be used within a constructor or method.