AP CS: Lesson 17.1: Object Class & Constructors

Name: _____ Period: ___

Object Class:

Class: A program entity that represents either:

1. A program / module, or
2. A template for a new type of objects.

A**bstraction**: A distancing between ideas and details. We can use Objects without knowing how they work, just like you don't need to know how an iPod's circuitry works just how to use it's control panel (methods) and the settings (fields).

Client Program (Class): A program that uses Objects
Object (Class): An entity that combines state and behavior.

1. **State**: Fields (i.e. int, double, array, other objects)
2. **Behavior**: Methods
   a. **mutator**: A method that modifies an object's state. Examples: setLocation, translate. Typically has a void return type
   b. **accessor**: A method that lets clients examine object state. Examples: distance, distanceFromOrigin. Typically has has a non-void return type

Object States: fields: A variable inside an object that is part of its state.

```
public class ObjectName {
        type name1;     // field1
        type name1;     // field2
    }
```

Creating an instance of an Object:

```
ObjectName objectInstance = new ObjectName();   // More on optinal parameters later
```

For example for Object `Point`:

```
Point p1 = new Point();
```

Accessing an Objects Fields:

access: `objectInstance.field`
modify: `objectInstance.field = value;`

For example for Object `Point`:

access: `p1.x`
modify: `p1.y = 6;`

Object Behavior: Methods:
Instance method (or **Object method**): Exists inside each object of a class and gives behavior to each object. NOTE: no "static" in declaration.

```
public type name(parameters) {
    statements;
}
```

AP CS: Lesson 17.1: Object Class & Constructors

Kinds of Methods:

- **mutator**: A method that modifies an object's state.
  - Examples: setLocation, translate
- **accessor**: A method that lets clients examine object state.
  - Examples: distance, distanceFromOrigin
  - often has a non-void return type

Implicit parameter: The object on which an instance method is called.

```
p1.draw(g);
```

the object referred to by p1 is the implicit parameter, the method will act on its (p1's) fields

The instance method can refer to it's Object's fields

Constructor (method): Initializes the state of new objects. (where type is the Object's name)

```
public type(parameters) {
    statements;
}
```

A class can have **multiple constructors**. But each one must accept a unique set of parameters. BTW, this is also true of methods in Object Classes and Client Programs.

AP CS: Lesson 17.1: Object Class & Constructors

Class Notes: