## Week 2 - Gapped Handout

# **Title: Defining Classes and Accessing Objects**

(Must be completed before attending tutorial 2 in week 3)

Reference text: Y Daniel Liang (2015). Introduction to Java programming. Boston: Pearson. Pg. 322. Ch 9.2

## Gap fillers (not in order)

interchangeable

properties

identity

arguments

no-arg

behaviors

objects

dot

instance

data

getArea()

class

template

return

state

operator

objects

behavior

signatures

getPerimeter()

dot

values

actions

blueprint

instantiation

data

actions

creation

identified

initialization

name

overloaded

void

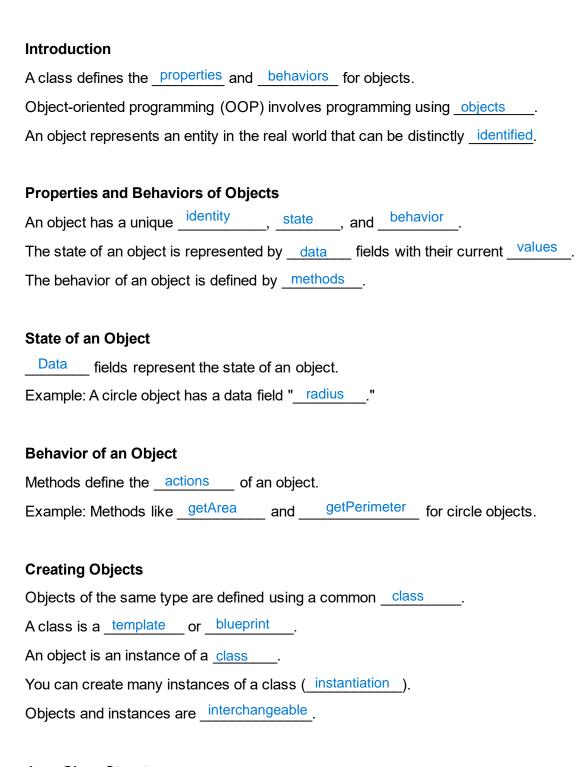
defined

no-argument

defined

instance

calling



### **Java Class Structure**

A Java class uses variables for <u>instance</u> fields and methods for <u>actions</u>.

Constructors are special methods used for object <u>creation</u> and <u>initialization</u>

#### **Constructors**

Constructors have the same <u>name</u> as the class.

Constructors do not have a <u>return</u> type.
Constructors are invoked using the new <u>operator</u> .
Constructors initialize <u>objects</u> .
Overloading Constructors
Constructors can be <u>overloaded</u> .
Multiple constructors can have the same name but different <u>signatures</u> .
Common Mistake
Avoid using the <u>void</u> keyword in front of a constructor.
Constructing Objects
• ,
To create an object, invoke a constructor using the new <u>operator</u> .
No-Argument Constructor
A class often provides a constructor without <u>arguments</u> .
It is referred to as a or constructor.  no-arg default
Default Constructor
If no constructors are explicitly defined, a default constructor is implicitly <u>defined</u> .
Accessing Objects via Reference Variables
Object members can be accessed using the <u>dot</u> operator.
Example: objectRefVardata references a data field in the object.
Example: objectRefVar.method (arguments) invokes a method on the object.
Instance Variables and Methods
Instance variables are specific to an <u>object</u> .

Instance methods can be invoked only on a specific	instance	
--	----------	--

## **Calling Object**

The object on which an instance method is invoked is called a <u>calling</u> object.

## **Key Points to Note**

- ✓ OOP involves programming with objects.
- ✓ Objects have state/attribute and behavior.
- ✓ Data fields represent object state, and methods define behavior.
- ✓ Objects are instances of classes.
- ✓ Constructors initialize objects.
- ✓ Overloaded constructors allow different initializations.
- ✓ Avoid using the void keyword in constructors.
- ✓ Access object members using the dot operator.
- ✓ Instance variables and methods are specific to instances.
- ✓ The calling object is the object on which a method is invoked.

It is good to refer below URLs as well.

- 1. W3schools (2020). Java OOP (Object-Oriented Programming). [online] W3schools.com. Available at: <a href="https://www.w3schools.com/java/java\_oop.asp">https://www.w3schools.com/java/java\_oop.asp</a>.
- 2. W3schools.com. (2019). Java Classes and Objects. [online] Available at: https://www.w3schools.com/java/java classes.asp.