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

Introduction

A class defines the properties and behaviors for objects.

Object-oriented programming (OOP) involves programming using objects.

An object represents an entity in the real world that can be distinctly identified.

Properties and Behaviors of Objects

An object has a unique identity, state, and behavior.

The state of an object is represented by data fields with their current values.

The behavior of an object is defined by methods.

State of an Object

Data fields represent the state of an object.

Example: A circle object has a data field "radius."

Behavior of an Object

Methods define the actions of an object.

Example: Methods like getArea and getPerimeter for circle objects.

Creating Objects

Objects of the same type are defined using a common class.

A class is a template or blueprint.

An object is an instance of a class.

You can create many instances of a class (instantiation).

Objects and instances are interchangeable.

Java Class Structure

A Java class uses variables for instance fields and methods for actions.

Constructors are special methods used for object creation and initialization.

Constructors

Constructors have the same name as the class.

Constructors do not have a _return_ type.

Constructors are invoked using the new _operator_.

Constructors initialize _objects_.

Overloading Constructors

Constructors can be _overloaded_.

Multiple constructors can have the same name but different _signatures_.

Common Mistake

Avoid using the _void_ keyword in front of a constructor.

Constructing Objects

To create an object, invoke a constructor using the new _operator_.

No-Argument Constructor

A class often provides a constructor without _arguments_.

It is referred to as a -_no-arg_ or -_default_ constructor.

Default Constructor

If no constructors are explicitly defined, a default constructor is implicitly _defined_.

Accessing Objects via Reference Variables

Object members can be accessed using the _dot_ operator.

Example: objectRefVar._data_ 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 _object_.

Instance methods can be invoked only on a specific _____instance_____.

Calling Object

The object on which an instance method is invoked is called a __calling____ 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: https://www.w3schools.com/java/java_oop.asp.
2. W3schools.com. (2019). Java Classes and Objects. [online] Available at: https://www.w3schools.com/java/java_classes.asp.