[4.1] Custom Classes

1. **What is a class and what is an object?**

ØA class is an entity that determines how an object will behave and what the object will contain. ØA class is a blueprint or set of instruction to build a specific type of object.

Objects

ØAn object is an instance of a class.

ØAn object can store data. The data stored in an object are commonly called fields.

**Class vs Object**

Class

Class is a collection of similar objects

Class is conceptual (is a template)

No memory is allocated for a class

Class can exist without any objects

Object

Object is an instance of a class

Object is real

Each object has its own memory

Objects can not exist without a class

1. **How are objects of a class created?**

Creating an Object/Instance

ØThe new keyword is used to create an instance of a class.

1. **Instance variables vs local variables**
2. **Instance variables in relation with objects**
3. **Creating objects and accessing their fields**

Accessing an Object’s Data and Methods?

ØAn Object’s member refer to its data fields and methods. After an object is created, its data can be accessed and its methods can be invoked using the dot operator (.)

1. **How can we print object to show the information of the object, not the hashcode?**
2. **Java file name in relation to the public class in the file**
3. **How is this used**

[4.2] Static members

1. **What are the static members?**
2. **What are the properties of something that is static?**
3. **Class objects vs static fields**
4. **Static block and execution flow**

Static block is a set of instructions that is run only once when a class is loaded into memory.

• A static block is also called a static initialization block.

1. **Static method signature**

If you declare any method as static, it is known as a static method.

A method that can be called using a class name, without creating an object.

Objects can also call static methods.

Static methods can not access instance methods and instance variables directly

In static methods we can not use the “this” keyword

1. **Static objects in other classes**

[4.3] Constructors

A special method that every class MUST have

It’s used when we create the objects of a class

We can use constructors to initialize the object’s instance variables.

Execution of a constructor ALWAYS depends on the object

1. **What is the purpose of the constructor in a class?**
2. **Constructor’s relation with new**
3. **The rules of creating constructors**

• Constructor is a special method that matches the name of the class and has no return type nor a specifier

1. **What is the default constructor and what happens if you create a constructor manually?**
2. **Constructor overloading**

we can have multiple constructors in a class by implementing method overloading

* No-arg constructor
* Constructor with int argument
* Constructor with String argument

1. **Constructor vs method**

• Constructor doesn’t have a return type and constructor’s name must be same as the class name.

o Constructor is called automatically when a new object is created. Constructor is invoked implicitly.

o The Java compiler provides a default constructor if we don’t have any constructor.

o Constructors are not inherited by child classes

• Method have a return and the method’s name may or not be same as the class name

o Method is invoked explicitly.

o Method is not provided by compiler in any case.

o Methods are inherited by child classes.

1. **Rules of constructor chaining using this()**

this: refers to the object instances

this. : to call instance variables or methods

this() :

Constructor call: this()

1. only a constructor can call another constructor

2. constructor can be overloaded. the only way to have more than one constructor in class

3. constructor cannot be called by constructor name. we need this()

4. constructor call MUST be at first step

5. one constructor can ONLY call one constructor

6. Constructor cannot call or contain itself