

•Chapter 2

Basic Object Oriented Programming Concepts

Review

- A Java program consists of a set of classes.
- The Java program must have a `main()` method from where it begins its execution.
- Variables defined in a class are called the instance variables.
- There are two types of casting: widening and narrowing.
- Variables are basic unit of storage.
- Each variable has a scope and lifetime.
- Arrays are used to store several items of same data type in consecutive memory locations.

Review Contd...

- Java provides different types of operators. They include:
 - Arithmetic
 - Bitwise
 - Relational
 - Logical
 - Conditional
 - Assignment
- Java supports the following programming constructs:
 - `if-else`
 - `switch`
 - `for`
 - `while`
 - `do-while`
- The three jump statements—`break`, `continue` and `return` helps to transfer control to another part of the program.

Objectives

- *Explain the Java Program Structure*
- *Design a Simple Class*
- *Create objects*
- *Explain the concept of methods in a class*
- *Implement constructors*
- *List the features of Inheritance*
- *Differentiate between Overloading and Overriding of methods*
- *Identify the access specifiers and method modifiers*

Java Program Structure

- Environment Information
- Class Declaration
- Tokens:
 - Identifiers
 - Keywords / Reserve words
 - Separators
 - Literals
 - Operators

Data Abstraction

- Data Abstraction is the process of identifying and grouping attributes and actions related to a particular entity as relevant to the application at hand
- Advantages:
 - It focuses on the problem
 - It identifies the essential characteristics and actions
 - It helps to eliminate unnecessary detail

Data Abstraction (Contd...)

Attributes	Actions
Name of the Customer	Accept name of the customer
Address of the Customer	Accept address of the customer
Model of the car bought	Accept the model of the car purchased
Salesman who sold the car	Accept the salesman name who sold the car
	Generate the bill

Class

- A Class defines an entity in terms of common characteristics and actions

Class Customer
Name of the customer
Address of the customer
Model of the car bought
Salesman's name who sold the car
Accept Name
Accept Address
Accept Model of the car purchased
Accept the name of the salesman who sold the car
Generate the bill

Class

- A class defines a new data type.
- Every time an instance of a class is created, an object is created.
- The object contains its own copy of each instance variable defined by the class.
- A dot operator (.) is used to access these variables.
- The dot operator links the name of the object with the name of an instance variable.

Object

- An object is an instance of class
 - Stephen
 - Boston
 - Opel Astra
 - Robin



Stephen is an Object of the class Customer

Object (Contd...)

- Attribute
 - Characteristic that describes an object
- Operation
 - Service that can be requested of an object
- Method
 - Specification of how the requested operation is carried out
- Message
 - Request for an operation

Declaring Objects

- When a new class is created, a new data type is created.
- Objects are declared to represent the class.
- Obtaining objects of a class is a two-step process. They are:
 - First, a variable of the class type has to be declared. The variable does not define an object. It is a variable that can refer to an object.
 - Second, an actual physical copy of the object must be acquired and assigned to that variable. It is done by using the `new` operator.
- The `new` operator dynamically allocates memory for an object and returns a reference to it.
- All class objects must be dynamically allocated.

Class vs. Object

- Class defines an entity, while an object is the actual entity
- Class is a conceptual model that defines all the characteristics and actions required of an object, while an object is a real model
- Class is a prototype of an object
- All objects belonging to the same class have the same characteristics and actions

Methods in Classes

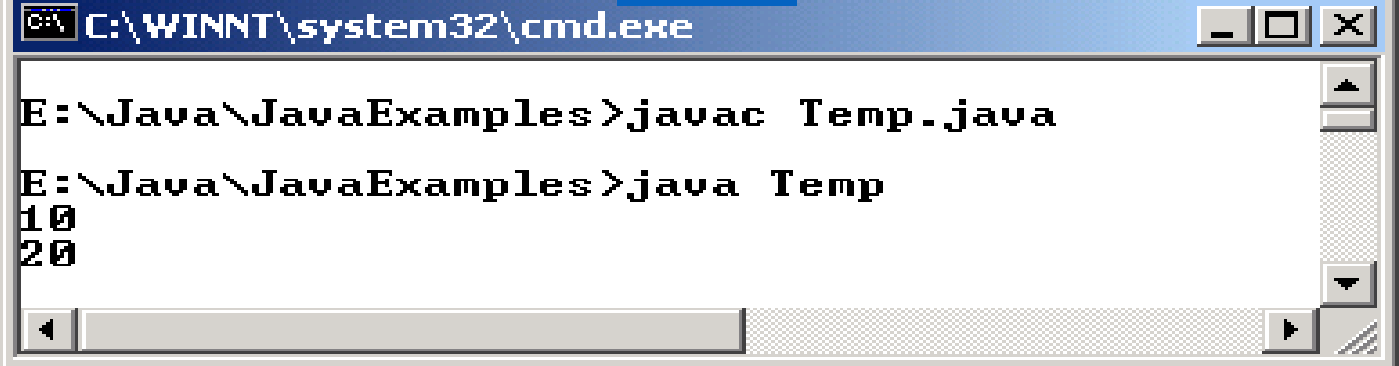
- A method is defined as the actual implementation of an operation on an object.
- Syntax

```
access_specifier modifier datatype method_name (parameter_list)
{
    //body of the method
}
```

- A method is always invoked relative to some objects of its class.
- Within a method there is no need to specify the object a second time.

Sample Usage of Method

```
class Temp
{
    static int num = 10;
    public void show()
    {
        System.out.println(num);
    }
    public static void main(String [] arg)
    {
        Temp tobj = new Temp();
        tobj.show();
        Temp t1Obj = new Temp();
        Temp.num = 20;
        t1Obj.show();
    }
}
```

Output

```
C:\WINNT\system32\cmd.exe

E:\Java\JavaExamples>javac Temp.java
E:\Java\JavaExamples>java Temp
10
20
```

Methods in Classes Contd...

- Methods that have a return type other than `void`, return a value to the calling routine using the `return` statement.
- Many methods need parameters.
- Parameters allow a method to be generalized.

Class Constructors

- Special methods are used to initialize member variables of the class.
- It has the same name as the Class name and does not have a return type.
- Called automatically and immediately after an object is created.
- Two types of constructors:
 - Parameterized constructors
 - Implicit or default constructors

Data Encapsulation

- The process of hiding the implementation details of an object from its user is called Encapsulation
- Advantages:
 - All attributes and methods required to accomplish a job can be created
 - Only those attributes and/or methods to be accessed by others can be made visible

Inheritance

- The attributes set for a class are inherited by the sub classes defined within the class.
- A class that is inherited from another class is called *subclass*.
- The class from which another class is derived is called *superclass*.
- *Subclass* is a specialized *superclass* and can access all the instance variables and methods defined by the *superclass*.
- To inherit a class, one has to use the keyword `extends` in the subclass.

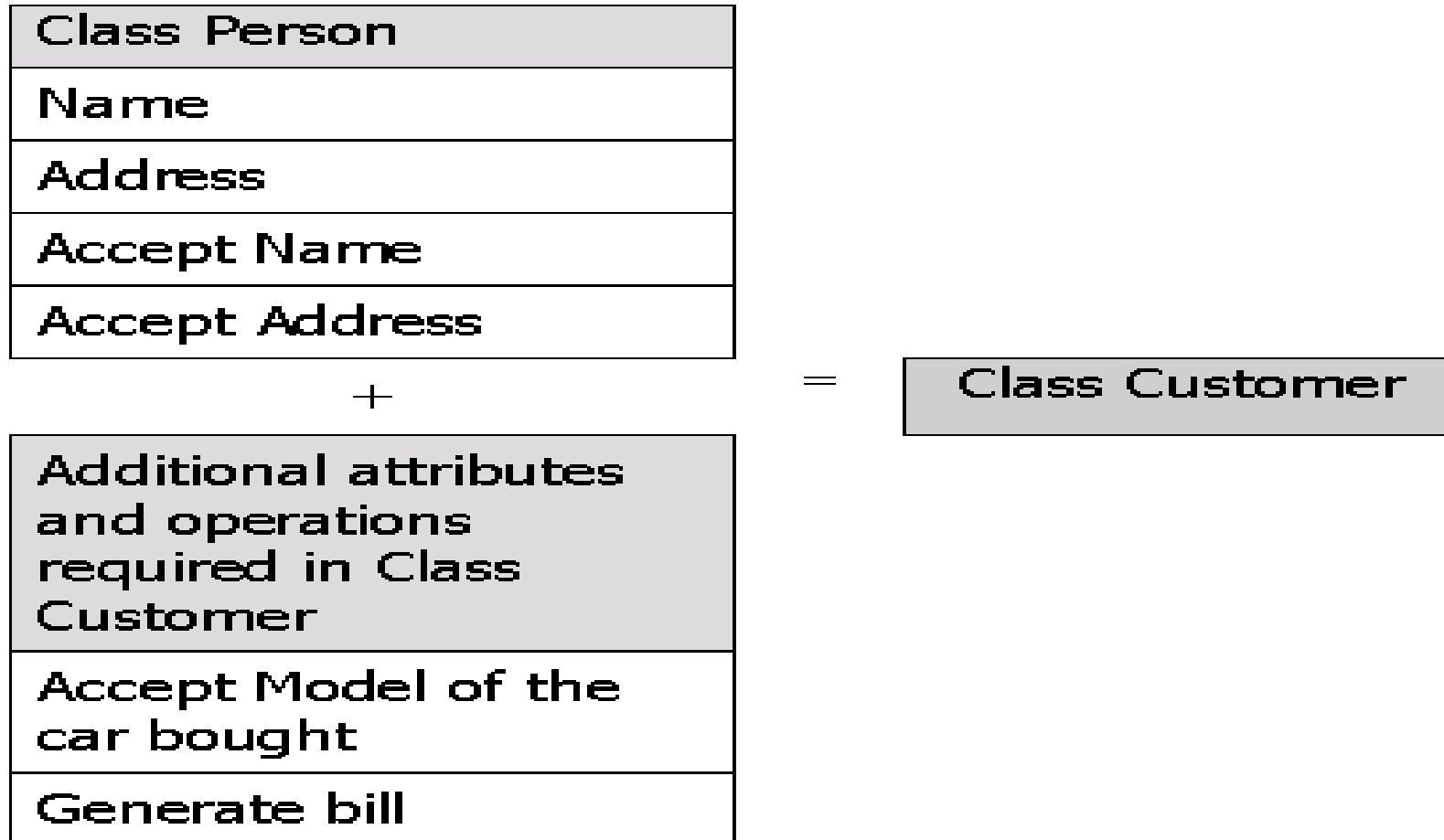
Inheritance

<i>Class Student</i>	<i>Class Employee</i>	<i>Class Customer</i>
Name	Name	Name
Address	Address	Address
Marks for subject 1	Salary	Model of the purchased car
Marks for subject 2	Designation	Accept name
Accept name	Accept name	Accept address
Accept address	Accept address	Accept Model
Accept marks	Accept Designation	Generate bill
Calculate total marks	Calculate salary	

Inheritance (Contd...)

<i>Class Person</i>
Name
Address
Accept name
Accept address

Inheritance (Contd...)



Derived Class constructors

- Has the same name as the subclass.
- Statement that calls the constructor of the superclass should be the first statement in the constructor of a subclass.
- The keyword `super` is used to call the superclass constructor.
- The keyword `super` can also be used to refer to methods or instance variable of the superclass.

Methods Overloading

- Characteristics of overloaded methods are:
 - Defined in the same class
 - Have the same name
 - Have different parameter lists
- Overloaded methods are a form of compile time polymorphism.

Methods Overriding

- Characteristics of overridden methods are:
 - Defined in the superclass as well as in the subclass.
 - Are redefined in the subclass.
- Overridden methods are a form of runtime polymorphism.

Access Specifiers for Methods

public

protected

private

Method Modifiers

- static
- final
- abstract
- native
- synchronized
- volatile

Nested Classes

- Defining one class within another is called 'Nesting'.
- Scope of a nested class is within the enclosing class.
- There are two types of nested class. They are:
 - Static
 - Non-static
- Inner Class is the non-static nested class.

Summary

- Import statements are used to access the Java packages required for the execution of the program.
- A token is the smallest unit in a program. There are five categories of tokens:
 - Identifiers
 - Keywords
 - Separators
 - Literals
 - Operators
- Class declaration only creates a template and not an actual object.
- Objects are instances of a class and have physical reality.
- Method is the actual implementation of an operation on an object.

Summary Contd...

- Constructors are used for automatic initialization of objects at the time of creation.
- The `super` keyword is used for calling the superclass constructors.
- To inherit a class from the superclass, the `extends` keyword is used.
- Overloaded methods are a form of static polymorphism and Overridden methods are a form of dynamic polymorphism.
- Java access specifiers: `public`, `protected`, `private` help in the implementation of encapsulation.
- The following modifiers are provided by Java: `static`, `final`, `abstract`, `native`, `synchronized`, and `volatile`.
- Nested class can be `static` or `non-static`.