



RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

BCS-503: Object Oriented Techniques

Lecture-11

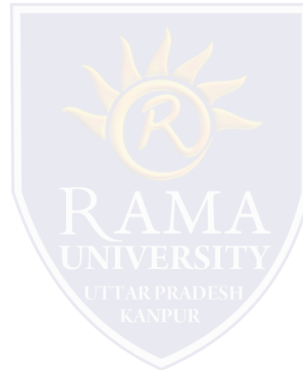
Preeti Singh

Computer Science & Engineering

OBJECTIVES

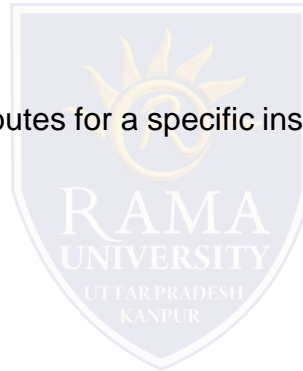
In this PPT, you will learn to:

- ❖ **Concepts of Static Modeling**
- ❖ **Finding classes and objects**
- ❖ **Class and Object Diagrams**



Static Modeling

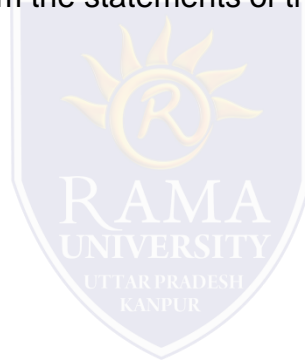
- Class diagrams to represent classes, interfaces, collaborations, and their relationships.
- Object diagrams to represent the values of attributes for a specific instance of a class and the relationship among objects.



FINDING CLASSES AND OBJECTS

You can derive classes and objects:

- From use cases of the software system.
- From responsibilities of the software system.
- By selecting all the nouns and noun phrases from the statements of the SRS document.



CLASS AND OBJECT DIAGRAMS

- Static Modeling technique is used to depict the static constituents of a software system.
- Class diagrams and Object diagrams are the Static Structure Models that are required to represent the static constituents of a software system.
- Class diagrams are used to represent the classes, attributes and the relationship between these classes whereas Object diagrams are used to represent the instance of these static constituents.

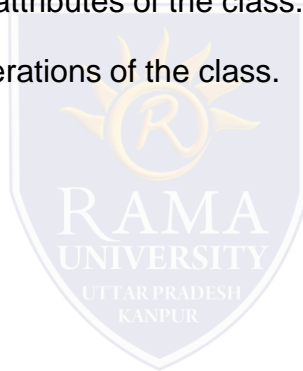


INTRODUCING CLASS DIAGRAMS

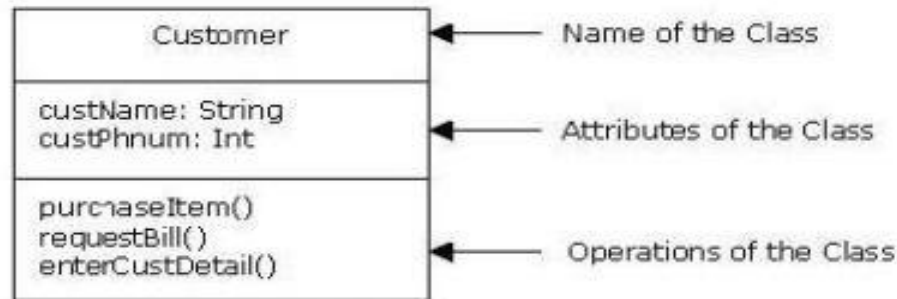
▪UML provides a class notation to represent classes in class diagrams.

The class notation has three compartments:

- 1) The first compartment depicts the name of the class.
- 2) The second compartment depicts the attributes of the class.
- 3) The third compartment depicts the operations of the class.

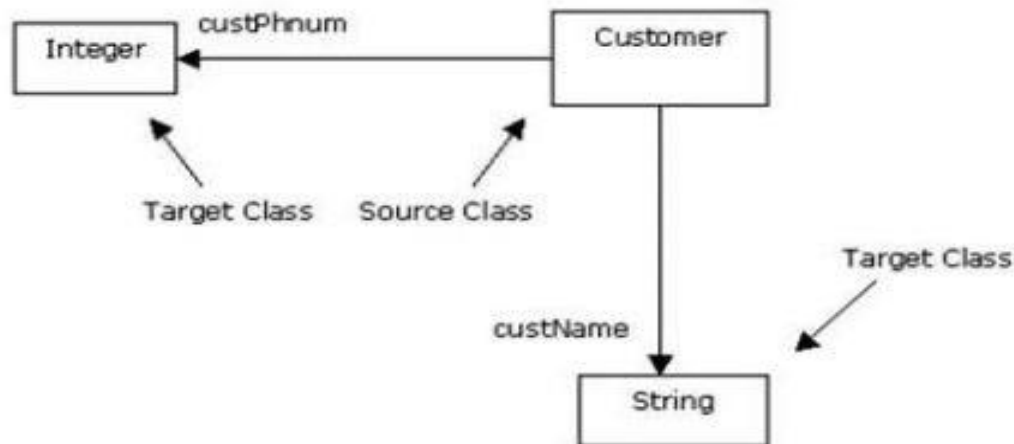


INTRODUCING CLASS DIAGRAMS (Contd.)



Class Notation

The preceding class notation depicts the attributes of a class as a string. You can also depict the attributes of a class as an association. For example, you can represent the properties of the `Customer` class, as shown in the following figure.



Association Notation

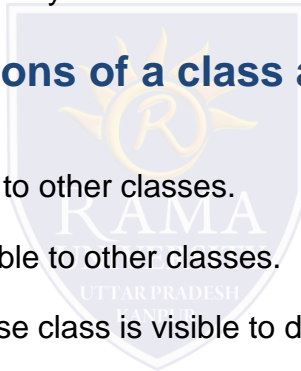
INTRODUCING CLASS DIAGRAMS (Contd.)

You can use class notation to:

- Analyze the requirements of the system. At the analysis level, the class notation depicts only the attributes and operations.
- Implement the requirements of the system at the implementation phase. At the implementation level, the class notation depicts attributes, operations, and their visibility.

Visibilities for attributes and operations of a class are:

- + : Indicates that the attribute or operation is visible to other classes.
- : Indicates that the attribute or operation is not visible to other classes.
- # : Indicates that the attribute or operation in the base class is visible to derived classes.
- ~: Indicates that the attribute or operation is visible to classes, which are within the same package.



INTRODUCING CLASS DIAGRAMS (Contd.)

Syntax for Attributes

The attributes of a class follow certain syntax in the class diagrams. The syntax to depict the attributes in a class diagram is:

```
visibility name: type multiplicity = default {property-string}
```

The preceding syntax contains the following fields:

- **visibility**: Specifies whether the attribute of the class is private, public, protected, or package.
- **name**: Indicates the name of the attribute. This is a mandatory field.
- **type**: Specifies the data type of the attribute.
- **multiplicity**: Indicates the number of values that can be stored in an attribute.
- **default**: Indicates the default value of the attributes of objects.
- **property-string**: Specifies property values that apply to the attribute, such as {ReadOnly} and {Ordered}. This is an optional field.

The following figure shows the example of the purchaseDate attribute of the Product class.

```
- purchaseDate: Date [1] = "01-01-2000" {readOnly}
```

↑ ↑ ↑ ↑ ↑ ↑

Visibility Name Type Multiplicity Default Property-String

An Example of an Attribute

INTRODUCING CLASS DIAGRAMS (Contd.)

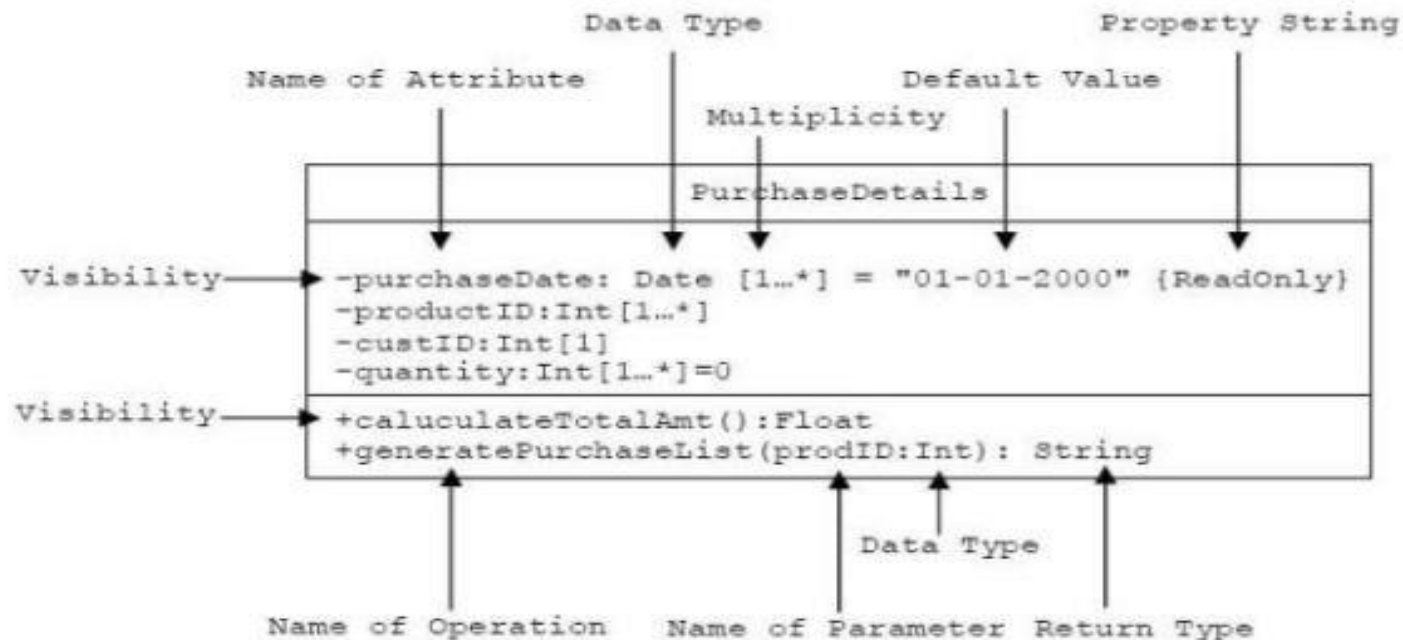
Syntax for Operations:

Syntax for representing operations in a class diagrams:

visibility name (parameter-list): return-type (property-string)

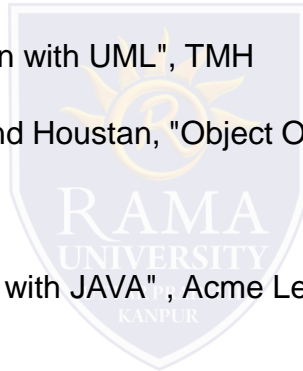
Example of a class:

The following figure shows a class notation with the syntax to depict attributes, operations, and parameters.



REFERENCES

1. James Rumbaugh et al, "Object Oriented Modeling and Design", PHI
2. Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide", Pearson Education
3. Naughton, Schildt, "The Complete Reference JAVA2", TMH
4. Mark Priestley "Practical Object-Oriented Design with UML", TMH
5. Booch, Maksimchuk, Engle, Young, Conallen and Houston, "Object Oriented Analysis and Design with Applications",
Pearson Education
6. Pandey, Tiwari, " Object Oriented Programming with JAVA" , Acme Learning
7. <https://www.javatpoint.com/java-tutorial>
8. <https://www.tutorialspoint.com/java/index.htm>
9. https://www.tutorialspoint.com/object_oriented_analysis_design/index.htm
10. <https://www.slideshare.net/niitstudentcare/>



MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q1. Which of the following visibilities provided by UML indicates that the attribute/operation in the base class are protected?

- a) +
- b) -
- c) #
- d) ~



MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q2. Which of the following is not needed to develop a system design from concept to detailed object-oriented design?

- a) Designing system architecture
- b) Developing design models
- c) Specifying interfaces
- d) Developing a debugging system



MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q3. Which of the following is a dynamic model that shows how the system interacts with its environment as it is used?

- a) system context model
- b) interaction model
- c) environmental model
- d) both system context and interaction

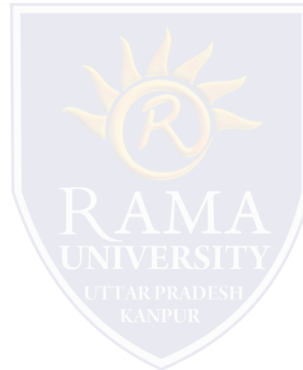


MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q4. Which of the following is a structural model that demonstrates the other systems in the environment of the system being developed?

- a) system context model
- b) interaction model
- c) environmental model
- d) both system context and interaction



MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q5. Which of the following come under system control?

- a) Reconfigure
- b) Shutdown
- c) Powersave
- d) All of the mentioned



In this PPT, you learned that:

- Static modeling involves identifying the static elements and their relationships to depict the static structure of the software system.
- The class and object diagrams are created to model the static structure of a software system.
- The class diagram notation represents the name, attributes, and operations of a class.
- The instances of the class diagram elements and their relationships need to be identified in order to create object diagrams. Instances refer to the objects of the software system.

