EXPERIMENT-8

AIM: Draw a UML Class diagram for any application.

DESCRIPTION:

A UML class diagram is made up of :

* A set of classes and
* A set of relationships between classes

A class is a description of a group of objects all with similar roles in the system, which consists of:

* Structural features (attributes) define what objects of the class "know"
  + Represent the state of an object of the class
  + Are descriptions of the structural or static features of a class
* Behavioral features (operations) define what objects of the class "can do"
  + Define the way in which objects may interact
  + Operations are descriptions of behavioral or dynamic features of a class

Class Notation

A class notation consists of three parts:

1. Class Name
   * The name of the class appears in the first partition.
2. Class Attributes
   * Attributes are shown in the second partition.
   * The attribute type is shown after the colon.
   * Attributes map onto member variables (data members) in code.
3. Class Operations (Methods)
   * Operations are shown in the third partition. They are services the class provides.
   * The return type of a method is shown after the colon at the end of the method signature.
   * The return type of method parameters are shown after the colon following the parameter name.
   * Operations map onto class methods in code

## Class Relationships

A class may be involved in one or more relationships with other classes. A relationship can be one of the following types:

1 . Association : A structural link between two peer classes.

2 . Dependency : Exists between two classes if changes to the definition of one may cause

changes to the other (but not the other way around).

3 . Generalization : Making two or more classes into a common class.

4 . Realization : Between class and interface.

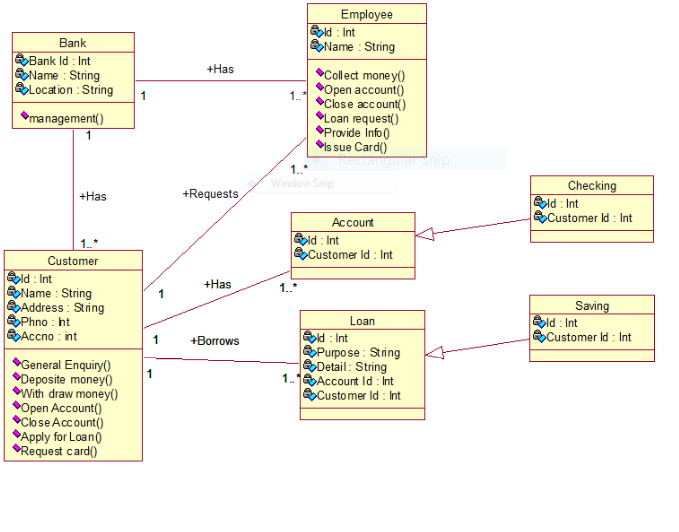
### Relationship - Roles

* A role is a directional purpose of an association.
* Roles are written at the ends of an association line and describe the purpose played by that class in the relationship.

## Multiplicity

How many objects of each class take part in the relationships and multiplicity can be expressed as:

* Exactly one - 1
* Zero or one - 0..1
* Many - 0..\* or \*
* One or more - 1..\*
* Exact Number - e.g. 3..4 or 6
* Or a complex relationship - e.g. 0..1, 3..4, 6.\* would mean any number of objects other than 2 or 5



SIGNATURE