International study centre

AN INTRODUCTION TO OBJECT-ORIENTATION AND THE JAVA PROGRAMMING LANGUAGE

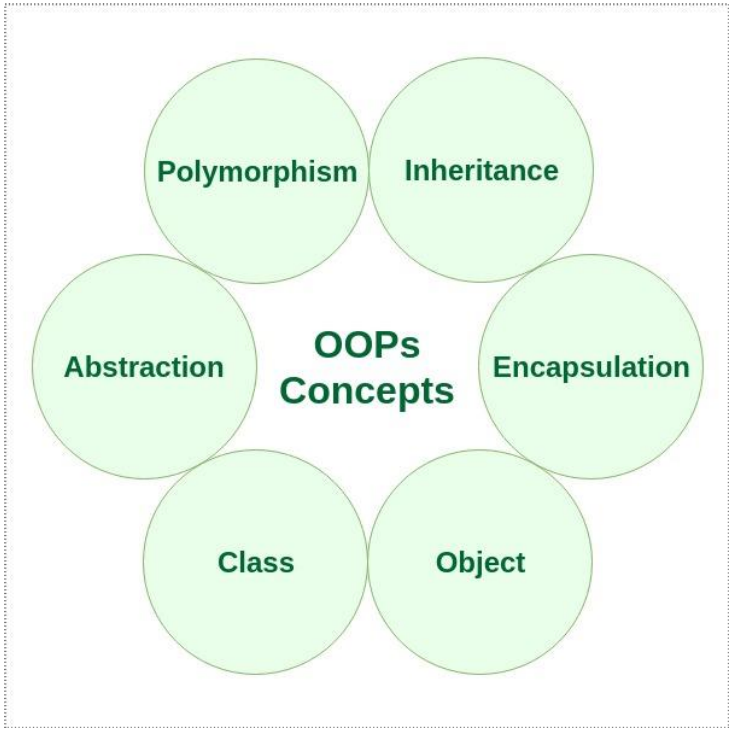
- Name : John Alamina
- Email : john.alamina@hud.ac.uk

Outline

- Object Oriented programming
- Abstraction, Encapsulation, Inheritance and polymorphism
- OOP keywords
- Class diagrams
- Class example
- Class members
- Object Instantiation
- Constructors
- this pointer
- String class

Object-Orientation.

- Object-orientation is a software design concept which is used to build systems from a collection of reusable components called objects.
- Objects contain data in the form of fields and functionality in the form of procedures. These are grouped together to represent an entity.
- Object-oriented design involves defining the objects and their interactions to solve a problem.



Features of Object-oriented Designs.

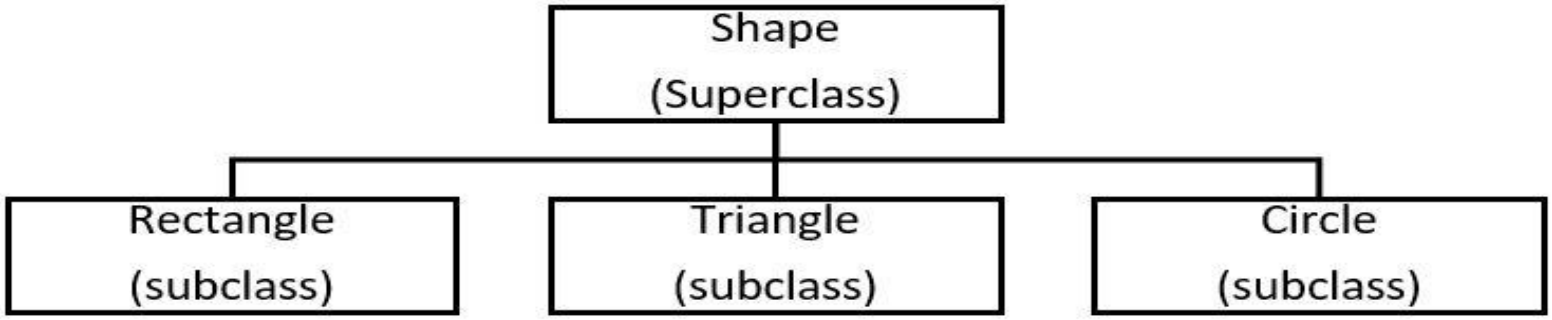
- **Abstraction** – The identification/description of the essential characteristics of an item.
- **Encapsulation** – The grouping of related concepts into one item, such as a class or component
- **Inheritance** – Inheritance enables new classes to receive the properties and methods of an existing class.
- **Polymorphism** – The ability of different object to respond to the same message in different ways.

Any Questions?

Terms Used in Object-oriented Design.

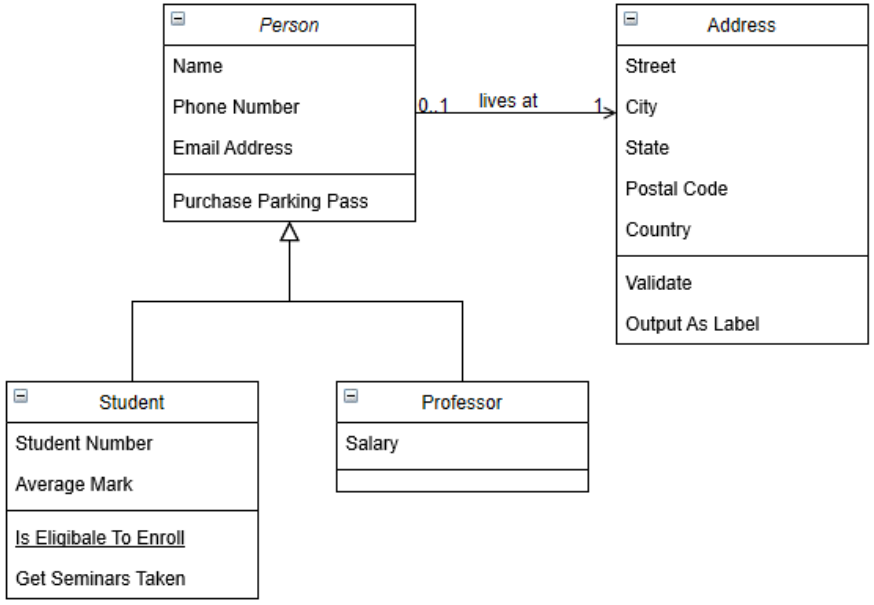
Object – An instance of a person, place, thing, event, concept, screen or report.

- **Class** – A template from which objects are created, it is a software ***abstraction*** of a group of *objects* with common properties (***Attributes***), behaviour (***operations***).
- **Subclass** – If class "B" inherits from class "A", class "B" is a subclass of "A" and "A" is a ***superclass*** of "B".
- **Abstract class** – A class that does not have objects created from it.
- **Association** – Relationship between objects.



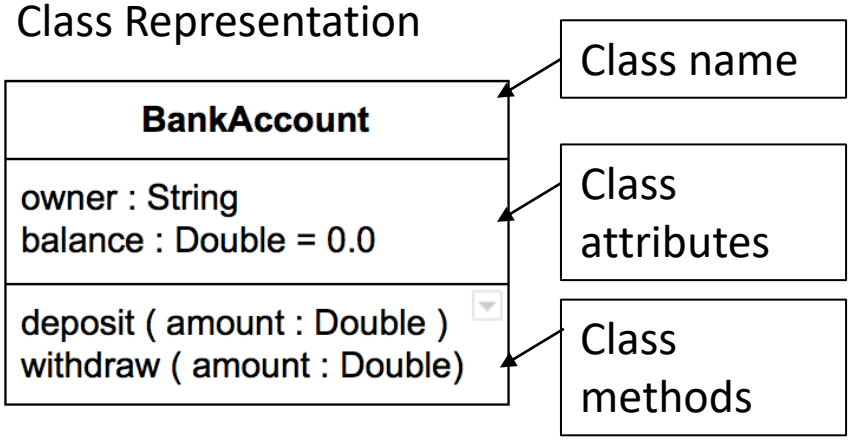
Working with class diagrams

- Go to www.draw.io >> choose “save diagrams to device” >> click “create new diagram” >> select “Class Diagram”.
- The following example should be shown on your screen:



Representations in UML.

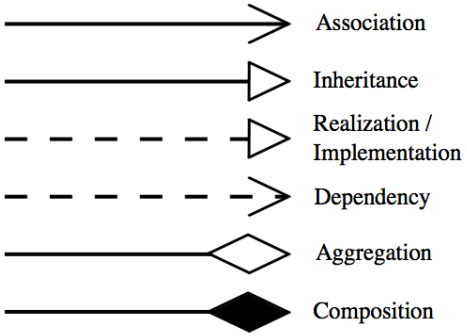
Class Representation



Visibility of class members

public	+	anywhere in the program and may be called by any object within the system
private	-	the class that defines it
protected	#	(a) the class that defines it or (b) a subclass of that class
package	~	instances of other classes within the same package

Relationships



BankAccount

-owner : String
-balance : Double = 0.0

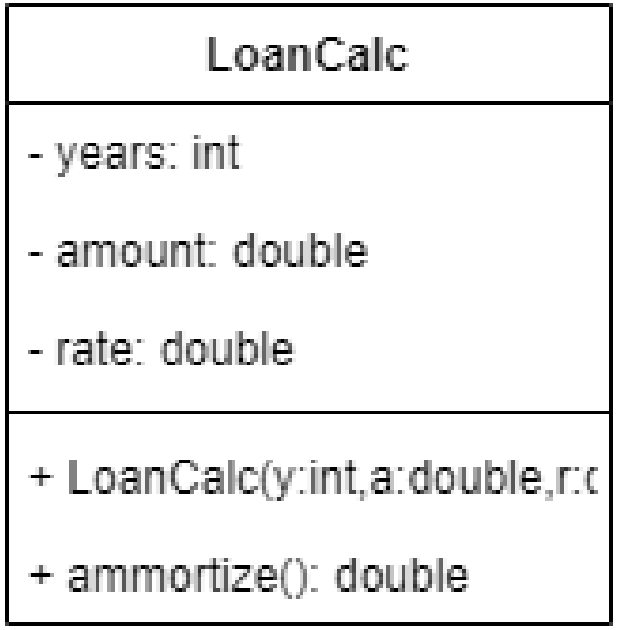
+deposit (amount : Double)
-withdraw (amount : Double)

Any Questions?

Loan Calculator Example

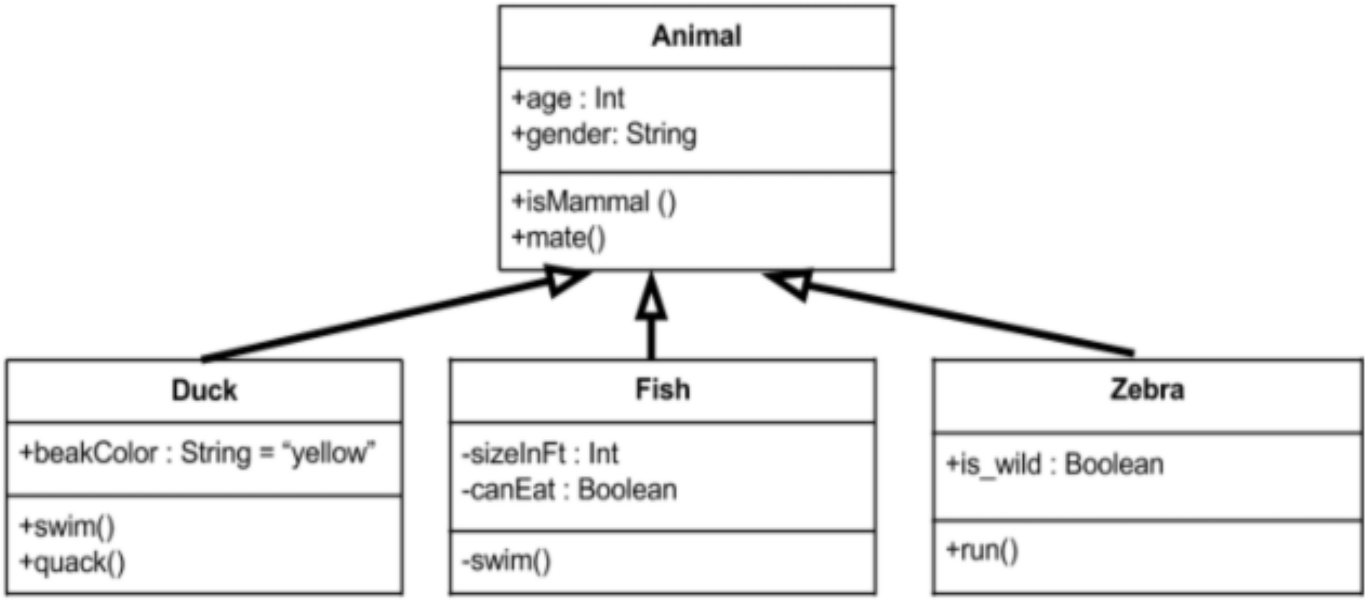
- Given the initial loan amount, an annual interest rate, and number of years,
- We can amortize (get monthly payments) using the following formula
- P=A/D, where P is the monthly payment, A is the initial amount and
- Discount Factor, D is
- $D = \frac{[(1+r)^n]-1}{r\times(1+r)^n}$
- where
  - n= number of payments to be made
  - r = interest rate per payment period

Loan Calculator Class diagram




```
public class LoanCalc{

   private int yrs;
   private double rate;
   private double amount;

   public LoanCalc(int y,double r, double a){
    this.rate=r;
    this.amount=a;
    this.yrs=y;
   }

   public double ammortize() {
      double r=rate/(100*12);
      double n=yrs*12;
      double a=amount;
      double d = (Math.pow((1 + r),n)- 1)/ (r*Math.pow((1 + r),n));
      return a/d ;
   }

   public static void main(String ar[]){
      int y=7;int r=3;double a=10000;
      LoanCalc carloan=new LoanCalc(y,r,a);
      double p=carloan.ammortize();
      System.out.println(String.format("monthly payments on %3.2f for %dyrs at %d%% is %3.2f",a,y,r,p));
   }
}
```

Inheritance Example



Animal is the super class. Duck, Fish, and Zebra are subclasses and they inherit all the members from the superclass. The subclasses also have their own attributes and methods.

The String Class

- Although string literals are the same in C++ and Java,
- There are two types of string variables in C++, the character array and the string object in the standard namespace.
- In Java the string variable is represented as a String object in the Java.lang package.
- Declaring and assigning a value to a string object.

```
String s;

s = “hello world!”;
```

- Declaring a string object using the constructor

```
String s = new String(“Hello world”);
```

String methods (operations)

- `s1.equals(s2)`
- `s1.equalsIgnoreCase(s2)`
- `s1.length()`
- `s1.charAt(N)`
- `s1.substring(N,M)`
- `s1.indexOf(s2)`
- `s1.compareTo(s2)`
- `s1.toUpperCase()`
- `s1.toLowerCase().`
- `s1.trim()`
- `s1.format()`

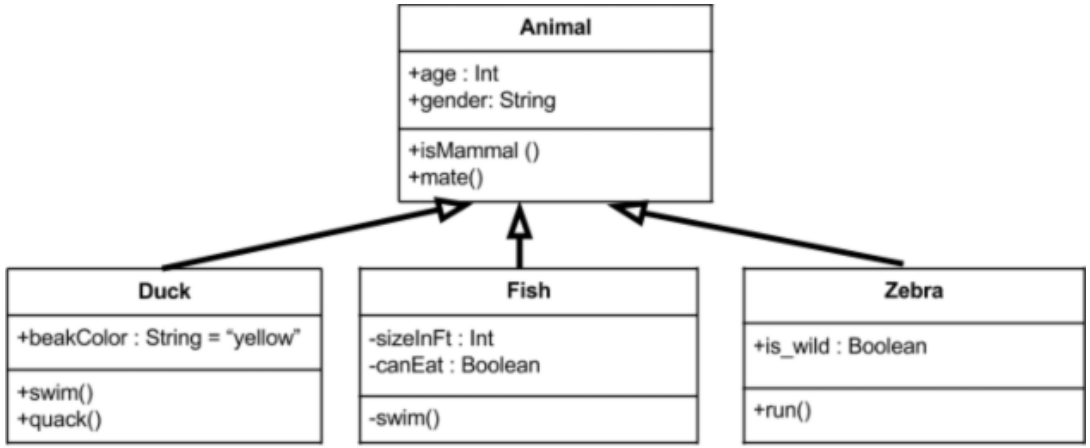
Any Questions?

Exercises

1. Write a Loan Calculator class given the initial loan amount, an annual interest rate, and number of years, amortization can be realised using the following formula P=A/D, where P is the monthly payment, A is the initial amount and Discount Factor, D is

$$D = \frac{[(1+r)^n] - 1}{r \times (1+r)^n}$$

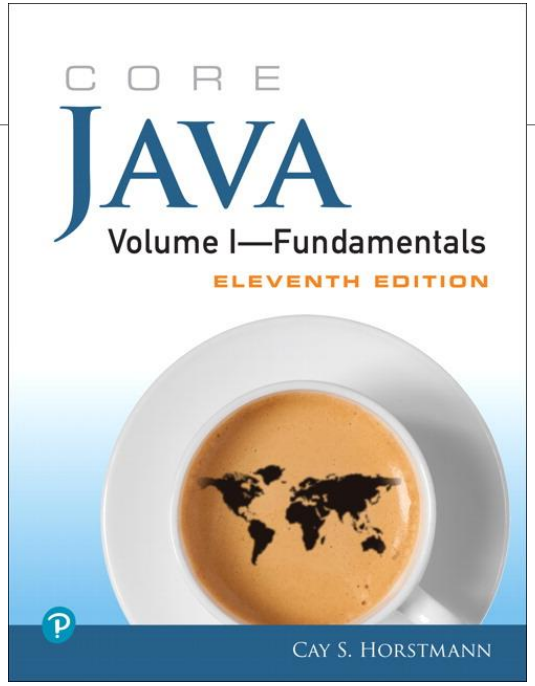
2. Implement the following class diagram in Java, making any necessary adjustments

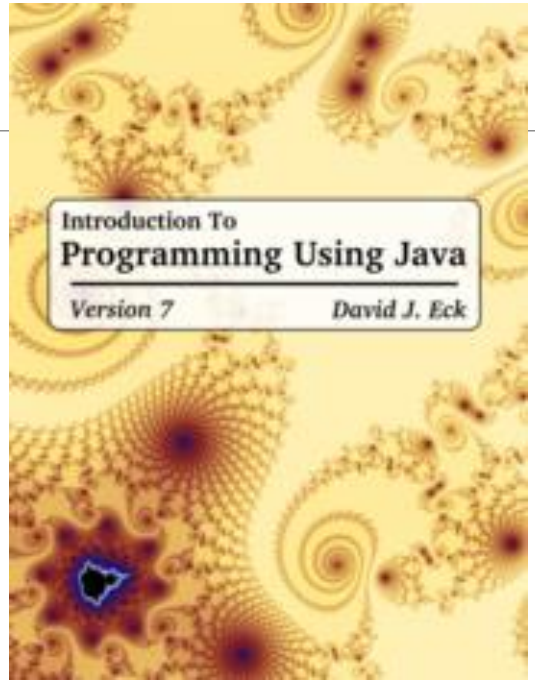


3. Write a program that can reverse any string e.g. "Hello World" becomes "dlroW olleH".

4. Write a program given the string "The quick brown fox jumped over the lazy dog" can count the number of spaces contained within the string.

Supplementary material





- The Java Tutorial
- Java API documentation
- Link to today's Session screencast
- Link to John's Group Padlet
- Link to Kelly's Group Padlet