Handout 9

Constructors

Constructor:

- Method that initializes values of new instances of a class.
  - Same name as the class
  - No return type

Default, no-arg constructor:

- If no constructor provided by the programmer, then the Java compiler automatically provides the constructor with no arguments.
- Once another constructor has been added, then the no-arg constructor must be explicitly specified.

Example:

```
BankAccount myAccount = new BankAccount();  // default constructor
myAccount.deposit(45);                      // add 45 to balance
```

Explicit constructors:

- Ability to have multiple constructors, each with the same name but different arguments, is another example of **method overloading**.

```
public class BankAccount {

        // instance variable
      private double balance;
      private String accountID;

    // Constructors:
    // no arg constructor.
    // balance defaults to 0, or can be set explicitly
      public  BankAccount (){ }

      //a 1-arg. Constructor.
      public  BankAccount (double b){
                  this.deposit(b);
            }

      //a 2-arg. Constructor.
      public  BankAccount (double b, String accNbr){
                  this.deposit(b);
                  this.setAccountID(accNbr);
            }
```

Using constructors

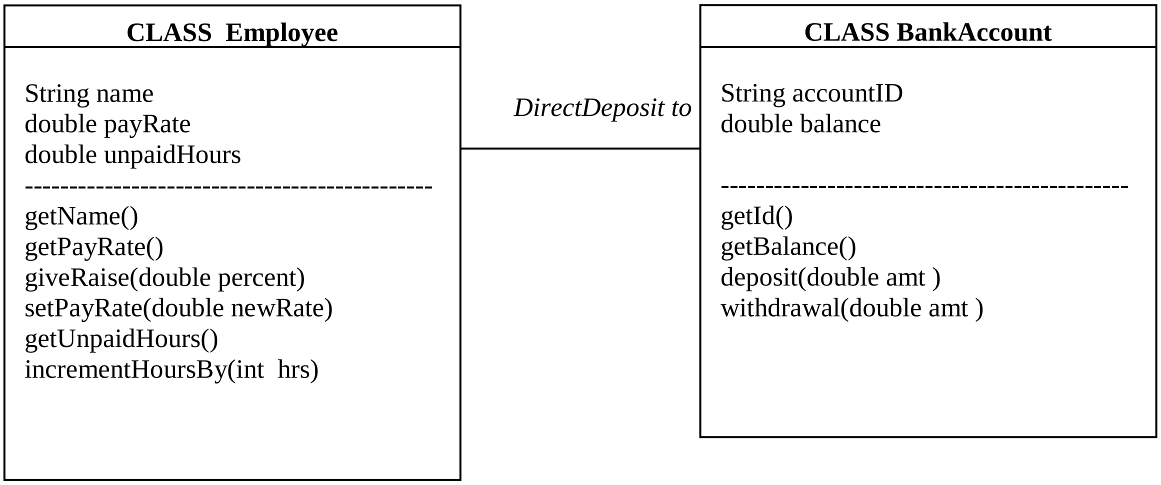
```
/* *  Example: three are three different constructors for BankAccount class
 */
public class BankAccountDemo {

      public static void main(String[] args) {

            BankAccount acc1 = new BankAccount();
            System.out.println(acc1);

            BankAccount acc2 = new BankAccount(100);
            System.out.println(acc2);

            BankAccount acc3 = new BankAccount(1000, "ADG-4553");
            System.out.println(acc3);

      }
}
```

Problem:

1. Add constructors to class Employee
   - with parameters defining name (String) and payRate (double).
   - a no-argument constructor

2. Implementing object associations:
   Applications typically consist of collections of objects of different related classes, working together.

Given class Employee and BankAccount, implement direct deposit as a form of payment to their employees. (The account owner may not be the same person as the employee)



We already have the separate class definitions for class Employee and BankAccount

To implement direct deposit feature, perform the following using the Employee and Bank Account class specifications:

a) Add a field of type BankAccount to each Employee object.
b) Add a new constructor to allow for specification of direct deposit account as well as the name and pay rate of an employee.
c) Add accessor and mutator methods for the direct deposit instance variable.
d) Add instance method ***pay*** to Employee class, which should work as follows:

   Calculate the amount due to employee for unpaidHours.

   In case employee does not have the direct deposit account specified on record (i.e. the value of the appropriate instance var is null), print a check for the specified amount.

   Otherwise, if direct deposit account field is not null, deposit the amount due to the employee on their direct deposit account (Hint: use instance method ***deposit*** of the BankAccount class).

   Reset the number of unpaidHours to 0.