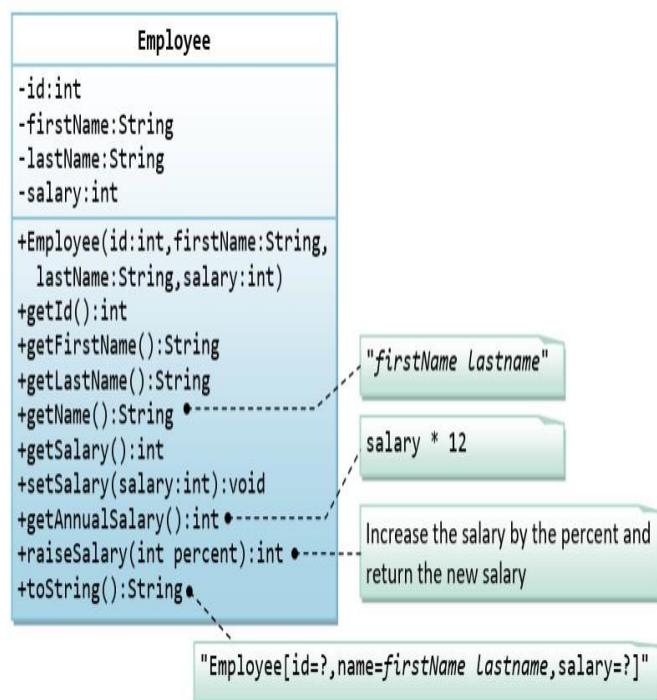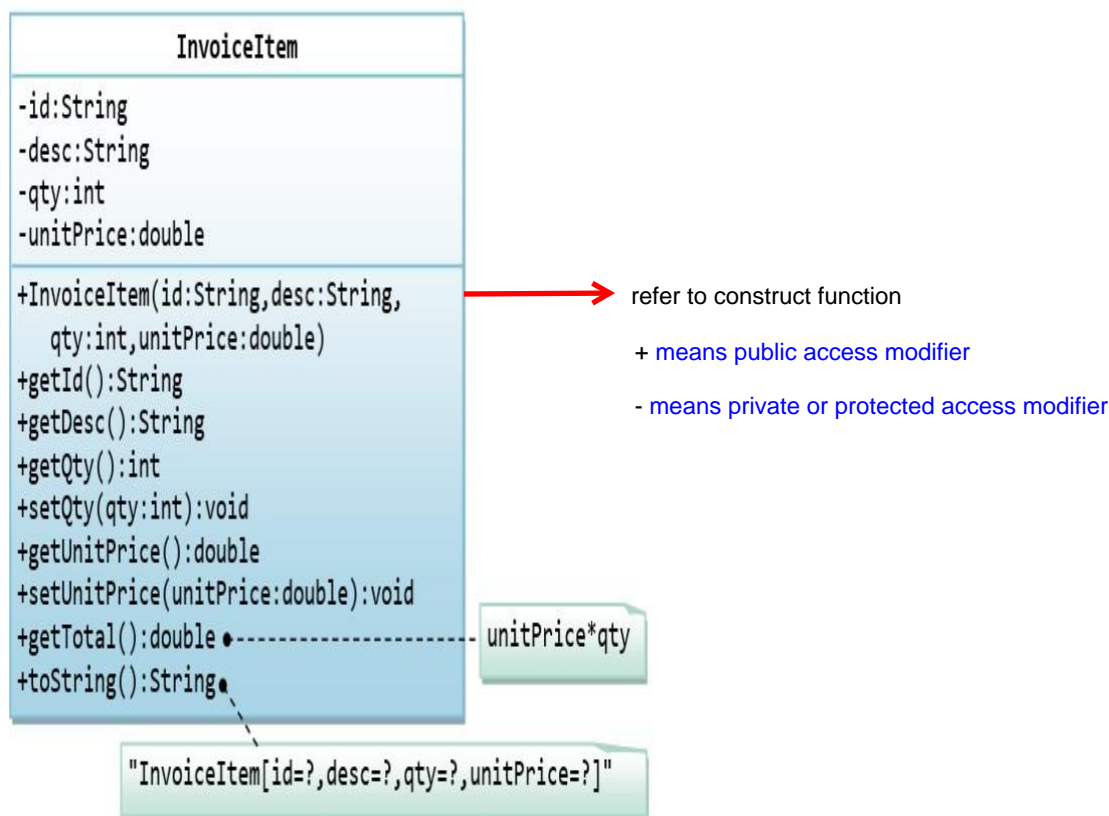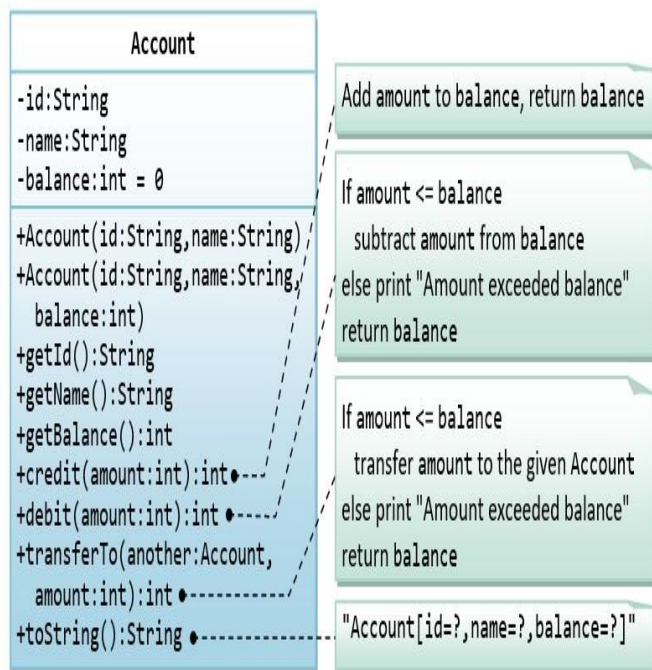A class called `Employee`, which models an employee with an ID, name and salary, is designed as shown in the following class diagram. The method `raiseSalary(percent)` increases the salary by the given percentage. Write the `Employee` class.

```
┌─────────────────────────────────────┐
│              Employee                │
├─────────────────────────────────────┤
│ -id:int                             │
│ -firstName:String                   │
│ -lastName:String                    │
│ -salary:int                         │
├─────────────────────────────────────┤
│ +Employee(id:int,firstName:String,  │
│    lastName:String,salary:int)      │
│ +getId():int                        │
│ +getFirstName():String              │
│ +getLastName():String               │
│ +getName():String                   │
│ +getSalary():int                    │
│ +setSalary(salary:int):void         │
│ +getAnnualSalary():int              │
│ +raiseSalary(int percent):int       │
│ +toString():String                  │
└─────────────────────────────────────┘
```

"firstName lastname"

salary * 12

Increase the salary by the percent and return the new salary

"Employee[id=?,name=firstName lastname,salary=?]"

A class called `InvoiceItem`, which models an item of an invoice, with ID, description, quantity and unit price, is designed as shown in the following class diagram. Write the `InvoiceItem` class.

```
┌─────────────────────────────────────┐
│              InvoiceItem             │
├─────────────────────────────────────┤
│ -id:String                          │
│ -desc:String                        │
│ -qty:int                            │
│ -unitPrice:double                   │
├─────────────────────────────────────┤
│ +InvoiceItem(id:String,desc:String, │
│    qty:int,unitPrice:double)        │
│ +getId():String                     │
│ +getDesc():String                   │
│ +getQty():int                       │
│ +setQty(qty:int):void               │
│ +getUnitPrice():double              │
│ +setUnitPrice(unitPrice:double):void│
│ +getTotal():double                  │
│ +toString():String                  │
└─────────────────────────────────────┘
```

refer to construct function

+ means public access modifier

- means private or protected access modifier

unitPrice*qty

"InvoiceItem[id=?,desc=?,qty=?,unitPrice=?]"

A class called Account, which models a bank account of a customer, is designed as shown in the following class diagram. The methods credit(amount) and debit(amount) add or subtract the given amount to the balance. The method transferTo(anotherAccount, amount) transfers the given amount from this Account to the given anotherAccount. Write the Account class.

Account

```
-id:String
-name:String
-balance:int = 0

+Account(id:String,name:String)
+Account(id:String,name:String,
   balance:int)
+getId():String
+getName():String
+getBalance():int
+credit(amount:int):int
+debit(amount:int):int
+transferTo(another:Account,
   amount:int):int
+toString():String
```

Add amount to balance, return balance

If amount <= balance
   subtract amount from balance
else print "Amount exceeded balance"
return balance

If amount <= balance
   transfer amount to the given Account
else print "Amount exceeded balance"
return balance

"Account[id=?,name=?,balance=?]"

Write the classes as shown in the following class diagram. Mark all the overridden methods with annotation @Override.

abstract class

Person

```
-name:String
-address:String

+Person(name:String,address:String)
+getName():String
+getAddress():String
+setAddress(address:String):void
```
abstract method `+toString():String`

"Person[name=?,address=?]"

extends △

Student

```
-program:String
-year:int
-fee:double

+Student(name:String,address:String,
   program:String,year:int,fee:double)
+getProgram():String
+setProgram(program:String):void
+getYear():int
+setYear(year:int):void
+getFee():double
+setFee(fee:double):void
+toString():String
```

"Student[Person[name=?,address=?],
program=?,year=?,fee=?]"

Staff

```
-school:String
-pay:double

+Staff(name:String,address:String,
   school:String,pay:double)
+getSchool():String
+setSchool(school:String):void
+getPay():double
+setPay(pay:double):void
+toString():String
```

"Staff[Person[name=?,address=?],
school=?,pay=?]"