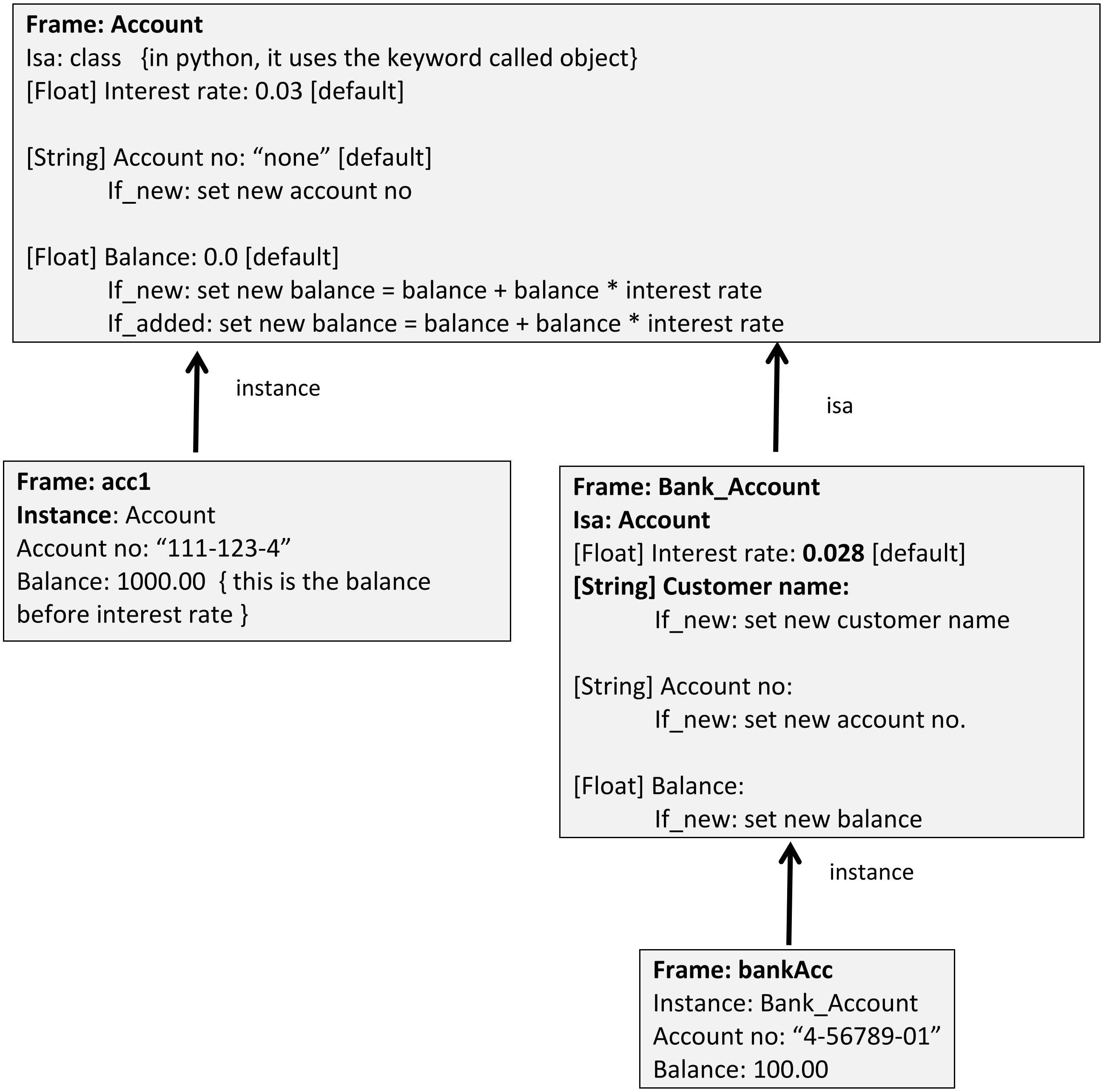


Frame: Account
Isa: class {in python, it uses the keyword called object}
[Float] Interest rate: 0.03 [default]

[String] Account no: “none” [default]
If_new: set new account no

[Float] Balance: 0.0 [default]
If_new: set new balance = balance + balance * interest rate
If_added: set new balance = balance + balance * interest rate

Frame: acc1
Instance: Account
Account no: “111-123-4”
Balance: 1000.00 { this is the balance before interest rate }

Frame: Bank_Account
Isa: Account
[Float] Interest rate: **0.028** [default]
[String] Customer name:
If_new: set new customer name

[String] Account no:
If_new: set new account no.

[Float] Balance:
If_new: set new balance

Frame: bankAcc
Instance: Bank_Account
Account no: “4-56789-01”
Balance: 100.00

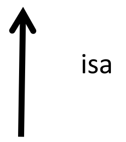
Python Code (inheritance)

```
class Account(object): # "object" is a keyword in python. Account is a
class
   interest_rate = 0.03 # interest_rate is an attribute or slot

   # a constructor with attributes that contain default values
    def __init__(self, account_no = "none", balance = 0.0):
        print "A new account is created"
        self.account_no = account_no
        self.balance = balance + balance * self.interest_rate

    def setBalance(self,bal):
        self.balance = bal + bal*self.interest_rate

    def __str__(self):  #to print the Account attributes in string
        return "The new balance %s is: %.2f" % (self.account_no,
self.balance)
```



```
class Bank_Account(Account): #Bank_Account is-a Account class

    interest_rate = 0.028 #interest rate is different from the parent
class

    #constructor with local attribute customer_name, two inherited
from the superclass
    def __init__(self, account_no, balance, customer_name):
        super(Bank_Account, self).__init__(account_no, balance)
        self.customer_name = customer_name

    def __str__(self): #to print the Bank_Account attributes in string
        return "Account No: %s\nCustomer Name: %s\nThe new balance
is: %.2f" % (self.account_no, self.customer_name, self.balance)
```

Python Code (instance from Account class)

```
class Account(object): # “object” is a keyword in python. Account is a
class
   interest_rate = 0.03 # interest_rate is an attribute or slot

   # a constructor with attributes that contain default values
    def __init__(self, account_no = "none", balance = 0.0):
        print "A new account is created"
        self.account_no = account_no
        self.balance = balance + balance * self.interest_rate

    def setBalance(self,bal):
        self.balance = bal + bal*self.interest_rate

    def __str__(self):  #to print the Account attributes in string
        return "The new balance %s is: %.2f" % (self.account_no,
self.balance)
```

instance

```
from account import Account  #import the Account class from the
account module

acc1 = Account() #acc1 is instantiated from the Account class

acc1.setBalance(1000) #add new balance to the slot balance

print "The interest rate: ", acc1.interest_rate #get the acc1
inherited interest rate

print acc1 #this invokes the __str__ method
```

Python Code (instance from Bank_Account class)

```
class Bank_Account(Account): #Bank_Account is-a Account class

    interest_rate = 0.028 #interest rate is different from the parent
class

    #constructor with local attribute customer_name, two inherited
from the superclass
    def __init__(self, account_no, balance, customer_name):
        super(Bank_Account, self).__init__(account_no, balance)
        self.customer_name = customer_name

    def __str__(self): #to print the Bank_Account attributes in string
        return "Account No: %s\nCustomer Name: %s\nThe new balance
is: %.2f" % (self.account_no, self.customer_name, self.balance)
```

↑ instance

```
account_no = raw_input("Enter the bank account no:")
balance = input("Enter the balance of the bank account:")
customer_name = raw_input("Enter the customer's name: ")

bankAcc = Bank_Account(account_no, balance, customer_name)
print bankAcc
```