```
class BankAccount {
  private int balance;
  private String accNum;

  public BankAccount(String a) {
    accNum = a;
    balance = 0;
  } //constructor

  public void deposit(int amount) {
    balance = balance + amount;
  } // deposit

  public int getBalance() {
    return balance;
  } // getBalance
} // BankAccount
```

```
class BankClient {
private BankAccount bk1
        = new BankAccount("a123");
...
public void cutDebt(int m) {
          int am1 = bk1.getBalance();
          if (am1 < 0)
                bk1.deposit(m);
} // cutDebt
} // BankClient
```

```
program BankAccountEx(input, output);
  type BankAccount =
    record
      balance: Integer;
      accNum: String;
    end;
  var bk1, bk2: BankAccount;
      am1: Integer;
  procedure mkBankAccount
   (var b: BankAccount; a: String);
  begin
    b.balance := 0;
    b.accNum := a;
  end {mkBankAccount};
```

```
  procedure deposit(
    var b: BankAccount; amount: Integer);
  begin
    b.balance := b.balance + amount;
  end {deposit};
  function getBalance(b:BankAccount):Integer;
  begin
    getBalance := b.balance;
  end {getBalance};
begin
  mkBankAccount(bk1, "1234");
  ... deposit(bk1, 6);
  ... am1 := getBalance(bk1); ...
end.
```

More abstract example

```
void ex(int par) { ... }

target.ex(actual);
```

C++ example

```
void swap(int& first, int& second) {
    int temp;
    temp = first;
    first = second;
    second = temp;
} // swap
```

With the call:

```
swap(dee, dum);
```

C example

```
void swap(int* first, int* second) {
        int temp;
        temp = *first;
        *first = *second; // what??
        *second = temp;
}
```

Method call:

```
swap(&dee, &dum);// recall & means address of
```

Object example

```
void update(Someclass f) {
    ...
    f.someChange(); ...
} // fun
```

and the method call:

```
target.update(p);
```

WhichMode example

from Comparative Programming Languages, p 137

```
var element: Integer;
a : array [1..2] of Integer;

procedure whichmode (x: ?mode Integer);
begin
a[1] := 6;
element := 2;
x := x+3;
end;

begin
a[1] := 1; a[2] := 2; element := 1;
whichmode (a[element]);
…
end.
```