print.py 22/2

```
    1.  #!/usr/bin/env python
    2.  # -*- coding: utf-8 -*-
    3.
#0  4.  class class1:    — no self ⇒ class1.func1() ⇒ TypeError: wrong
    5.      def func1():                                      # of args.
    6.          """ basic print function of class - no self arg """
    7.          print "class1.func1()"
    8.
    9.      def func2(self):
   10.  #0      """ basic print function of class - with self arg """
   11.          print "class1.func2()"
   12.
   13.      def func3(self):
   14.   #1     """ basic function of class - return constant value """
   15.          return 4
   16.
#1 17.  class class2(class1):
   18.      """ ~~test~~ class inheritance - class2 -> class1 """
   19.      pass
   20.
   21.  class class3(object):
   22.      """ class object """
   23.      def func1(self, a):
   24.          """ single argument function """
   25.          print a
   26.          return a
   27.
   28.      def func2(self, a, b=5):
   29.          """ two argument function, with second default(=5) argument """
   30.          return a + b + 60
   31.
   32.  class class4():
   33.      """ non-inheritance class """
   34.      def __init__(self, a, b):
   35.          """ multiple params class constructor """
   36.          self.a = a
   37.          self.b = b
   38.
   39.      def __init__(self, a, b, c):
   40.          """ (overloaded) multiple params class constructor """
   41.          self.a = a
   42.          self.b = b
   43.          self.c = c
   44.
#2 45.  class BankAccount(object):
   46.      """ Real trivial toy example """
   47.      CLASS_ATTRIBUTE = 'some value'
   48.      def __init__(self, initial_balance=0):
   49.          self.balance = initial_balance
   50.      def deposit(self, amount):
   51.          self.balance += amount
   52.      def withdraw(self, amount):
   53.          self.balance -= amount
   54.      def overdrawn(self):
   55.          return self.balance < 0
   56.      def __str__(self):
   57.          return '%s(%d)' % (self.__class__.__name__, self.balance)
   58.
   59.  def foo(class1, class2, n, d='default', *vargs, **kwargs):
   60.      """ lazy instantiation test for global function """
```

```
61 → 61.      class2.func2() → AttrError
62 → 62.      class1.func3()
63 → 63.      bank_account = BankAccount(n)
     64.  ~~#    class1.func1()              # incorrect function~~     ] comment line# stripped in
     65.  ~~#    bank_account = BankAccount() # purposely leave out param~~ ]  pyc file.
64 → 66.      print bank_account
65 → 67.      return n~~+3~~
     68.
     69.  def foo1(class1, class2, n, d='default', *vargs, **kwargs):
     70.      class2.func2()
     71.      class1.func1()
     72.      class1.func3()
     73.      bank_account = BankAccount(n)
     74.      print bank_account
     75.
     76.  def foo2(class1, class2, n, d='default', *vargs, **kwargs):
     77.      bank_account = BankAccount()     # purposely leave out param
     78.      print bank_account
     79.
     80.  # foo(class1(), class2(), 5)
     81.  if __name__ == "__main__":
     82.      my_account = BankAccount(15)
     83.      my_account.withdraw(5)
     84.      print my_account.balance
```

bytecode is LSB first (low endian)

```
Disassembly of `foo` (instrumented):
        org line#                                param#
1       2000   bytecode offset  0 LOAD_FAST      1 (class2) ✓
2       2003                    3 LOAD_ATTR      0 (func2) ✓
                                                             ← Exception
3       2006                    6 CALL_FUNCTION  0   field attr#
4       2009                    9 POP_TOP ✓
5       3000                   10 LOAD_FAST      0 (class1) ✓
6       3003                   13 LOAD_ATTR      1 (func3) ✓
7       3006                   16 CALL_FUNCTION  0           ← Exception .
8       3009                   19 POP_TOP
                                                 class order # (only used/referred)
9       4000                   20 LOAD_GLOBAL    2 (BankAccount)   in params .
10      4003                   23 LOAD_FAST      2 (n)
> 11    4006                   26 CALL_FUNCTION  1
12      4009                   29 STORE_FAST     6 (bank_account)
13      5000                   32 LOAD_FAST      6 (bank_account)
                                                 Why?
14      5003                   35 PRINT_ITEM
15      5004                   36 PRINT_NEWLINE
16      5005                   37 LOAD_CONST     1 (None)
17      5008                   40 RETURN_VALUE        if return'
(100%)
```

return n: 6000 37 LOAD_FAST

2(n)

↑

param_no.

- #0 class1
- #1 class2
- #2 n
- #3 d

6.1. Exception hierarchy

The class hierarchy for built-in exceptions is:

```
BaseException
 +-- SystemExit
 +-- KeyboardInterrupt
 +-- GeneratorExit
 +-- Exception
      +-- StopIteration
      +-- StandardError
      |    +-- BufferError
      |    +-- ArithmeticError
      |    |    +-- FloatingPointError
      |    |    +-- OverflowError
      |    |    +-- ZeroDivisionError
      |    +-- AssertionError
      |    +-- AttributeError
      |    +-- EnvironmentError
      |    |    +-- IOError
      |    |    +-- OSError
      |    |         +-- WindowsError (Windows)
      |    |         +-- VMSError (VMS)
      |    +-- EOFError
      |    +-- ImportError
      |    +-- LookupError
      |    |    +-- IndexError
      |    |    +-- KeyError
      |    +-- MemoryError
      |    +-- NameError
      |    |    +-- UnboundLocalError
      |    +-- ReferenceError
      |    +-- RuntimeError
      |    |    +-- NotImplementedError
      |    +-- SyntaxError
      |    |    +-- IndentationError
      |    |         +-- TabError
      |    +-- SystemError
      |    +-- TypeError
      |    +-- ValueError
      |         +-- UnicodeError
      |              +-- UnicodeDecodeError
      |              +-- UnicodeEncodeError
      |              +-- UnicodeTranslateError
      +-- Warning
           +-- DeprecationWarning
           +-- PendingDeprecationWarning
           +-- RuntimeWarning
           +-- SyntaxWarning
           +-- UserWarning
           +-- FutureWarning
       +-- ImportWarning
       +-- UnicodeWarning
       +-- BytesWarning
```

inspect_pyc <.pyc>

to_hexstr
unpack_pyc
show_consts
show_bytecode
show_code
show_file

~~introspection~~.py
prototype

in bytecode importmodule
① Class definitions < classes, methods] args.
② Execute function(s)
③ Construct objects

reflection.py

INSPECT!

inspect.is_xxx
abstract
builtin
class
code
datadescriptor
frame
function
generator
generatorfunction
getset descriptor
memberdescriptor
method
methoddescriptor
module
routine
traceback

Python ~~Byte~~Opcodes

```
def_op('STOP_CODE', 0)
def_op('POP_TOP', 1)
def_op('ROT_TWO', 2)
def_op('ROT_THREE', 3)
def_op('DUP_TOP', 4)
def_op('ROT_FOUR', 5)
def_op('NOP', 9)
def_op('UNARY_POSITIVE', 10)
def_op('UNARY_NEGATIVE', 11)
def_op('UNARY_NOT', 12)
def_op('UNARY_CONVERT', 13)
def_op('UNARY_INVERT', 15)
def_op('BINARY_POWER', 19)
def_op('BINARY_MULTIPLY', 20)
def_op('BINARY_DIVIDE', 21)
def_op('BINARY_MODULO', 22)
def_op('BINARY_ADD', 23)
def_op('BINARY_SUBTRACT', 24)
def_op('BINARY_SUBSCR', 25)
def_op('BINARY_FLOOR_DIVIDE', 26)
def_op('BINARY_TRUE_DIVIDE', 27)
def_op('INPLACE_FLOOR_DIVIDE', 28)
def_op('INPLACE_TRUE_DIVIDE', 29)
def_op('SLICE+0', 30)
def_op('SLICE+1', 31)
def_op('SLICE+2', 32)
def_op('SLICE+3', 33)
def_op('STORE_SLICE+0', 40)
def_op('STORE_SLICE+1', 41)
def_op('STORE_SLICE+2', 42)
def_op('STORE_SLICE+3', 43)
def_op('DELETE_SLICE+0', 50)
def_op('DELETE_SLICE+1', 51)
def_op('DELETE_SLICE+2', 52)
def_op('DELETE_SLICE+3', 53)
def_op('STORE_MAP', 54)
def_op('INPLACE_ADD', 55)
def_op('INPLACE_SUBTRACT', 56)
def_op('INPLACE_MULTIPLY', 57)
def_op('INPLACE_DIVIDE', 58)
def_op('INPLACE_MODULO', 59)
def_op('STORE_SUBSCR', 60)
def_op('DELETE_SUBSCR', 61)
def_op('BINARY_LSHIFT', 62)
def_op('BINARY_RSHIFT', 63)
def_op('BINARY_AND', 64)
def_op('BINARY_XOR', 65)
def_op('BINARY_OR', 66)
def_op('INPLACE_POWER', 67)
def_op('GET_ITER', 68)
def_op('PRINT_EXPR', 70)
def_op('PRINT_ITEM', 71)
def_op('PRINT_NEWLINE', 72)
def_op('PRINT_ITEM_TO', 73)
def_op('PRINT_NEWLINE_TO', 74)
def_op('INPLACE_LSHIFT', 75)
def_op('INPLACE_RSHIFT', 76)
def_op('INPLACE_AND', 77)
def_op('INPLACE_XOR', 78)
def_op('INPLACE_OR', 79)
def_op('BREAK_LOOP', 80)
def_op('WITH_CLEANUP', 81)
def_op('LOAD_LOCALS', 82)
def_op('RETURN_VALUE', 83)
```

```
def_op('IMPORT_STAR', 84)
def_op('EXEC_STMT', 85)
def_op('YIELD_VALUE', 86)
def_op('POP_BLOCK', 87)
def_op('END_FINALLY', 88)
def_op('BUILD_CLASS', 89)
HAVE_ARGUMENT = 90              # Opcodes from here have an argument:
name_op('STORE_NAME', 90)       # Index in name list
name_op('DELETE_NAME', 91)      # ""
def_op('UNPACK_SEQUENCE', 92)   # Number of tuple items
jrel_op('FOR_ITER', 93)
def_op('LIST_APPEND', 94)
name_op('STORE_ATTR', 95)       # Index in name list
name_op('DELETE_ATTR', 96)      # ""
name_op('STORE_GLOBAL', 97)     # ""
name_op('DELETE_GLOBAL', 98)    # ""
def_op('DUP_TOPX', 99)          # number of items to duplicate
def_op('LOAD_CONST', 100)       # Index in const list
name_op('LOAD_NAME', 101)       # Index in name list
def_op('BUILD_TUPLE', 102)      # Number of tuple items
def_op('BUILD_LIST', 103)       # Number of list items
def_op('BUILD_SET', 104)        # Number of set items
def_op('BUILD_MAP', 105)        # Number of dict entries (upto 255)
name_op('LOAD_ATTR', 106)       # Index in name list
def_op('COMPARE_OP', 107)       # Comparison operator
name_op('IMPORT_NAME', 108)     # Index in name list
name_op('IMPORT_FROM', 109)     # Index in name list
jrel_op('JUMP_FORWARD', 110)    # Number of bytes to skip
jabs_op('JUMP_IF_FALSE_OR_POP', 111) # Target byte offset from beginning of code
jabs_op('JUMP_IF_TRUE_OR_POP', 112)  # ""
jabs_op('JUMP_ABSOLUTE', 113)        # ""
jabs_op('POP_JUMP_IF_FALSE', 114)    # ""
jabs_op('POP_JUMP_IF_TRUE', 115)     # ""
name_op('LOAD_GLOBAL', 116)     # Index in name list
jabs_op('CONTINUE_LOOP', 119)   # Target address
jrel_op('SETUP_LOOP', 120)      # Distance to target address
jrel_op('SETUP_EXCEPT', 121)    # ""
jrel_op('SETUP_FINALLY', 122)   # ""
def_op('LOAD_FAST', 124)        # Local variable number
def_op('STORE_FAST', 125)       # Local variable number
def_op('DELETE_FAST', 126)      # Local variable number
def_op('RAISE_VARARGS', 130)    # Number of raise arguments (1, 2, or 3)
def_op('CALL_FUNCTION', 131)    # #args + (#kwargs << 8)
def_op('MAKE_FUNCTION', 132)    # Number of args with default values
def_op('BUILD_SLICE', 133)      # Number of items
def_op('MAKE_CLOSURE', 134)
def_op('LOAD_CLOSURE', 135)
def_op('LOAD_DEREF', 136)
def_op('STORE_DEREF', 137)
def_op('CALL_FUNCTION_VAR', 140)     # #args + (#kwargs << 8)
def_op('CALL_FUNCTION_KW', 141)      # #args + (#kwargs << 8)
def_op('CALL_FUNCTION_VAR_KW', 142)  # #args + (#kwargs << 8)
jrel_op('SETUP_WITH', 143)
def_op('EXTENDED_ARG', 145)
EXTENDED_ARG = 145
def_op('SET_ADD', 146)
def_op('MAP_ADD', 147)
```

...256

inspect_pyc.py

```
filename:     2012/simple_programs/12.pyc
magic number: 0x(03 f3 0d 0a)
timestamp:    1328114264 (Wed Feb  1 16:37:44 2012)
code
   co_argcount:        0
   co_cellvars:        ()
   co_filename:        './12.py'
   co_firstlineno:     1
   co_flags:           0x00040
   co_freevars:        ()
   co_lnotab:          '\x16\t\x0c\x01\r\x01'
   co_name:            '<module>'
   co_names:           ('object', 'BankAccount', 'my_account', 'withdraw',
'balance')
   co_nlocals:         0
   co_stacksize:       3
   co_varnames:        ()
   co_consts
      0 'BankAccount'
      1 (code object)
         co_argcount:        0
         co_cellvars:        ()
         co_filename:        './12.py'
         co_firstlineno:     1
         co_flags:           0x00042
         co_freevars:        ()
         co_lnotab:          '\x06\x01\x0c\x02\t\x02\t\x02'
         co_name:            'BankAccount'
         co_names:           ('__name__', '__module__', '__init__', 'deposit',
'withdraw', 'overdrawn')
         co_nlocals:         0
         co_stacksize:       2
         co_varnames:        ()
         co_consts
            0 0
            1 (code object)
               co_argcount:        2
               co_cellvars:        ()
               co_filename:        './12.py'
               co_firstlineno:     2
               co_flags:           0x00043
               co_freevars:        ()
               co_lnotab:          '\x00\x01'
               co_name:            '__init__'
               co_names:           ('balance',)
               co_nlocals:         2
               co_stacksize:       2
               co_varnames:        ('self', 'initial_balance')
               co_consts
                  0 None
               co_code
                  7c 01 00 7c 00 00 5f 00 00 64 00 00 53
                  disassembled:
                    3           0 LOAD_FAST                 1 (initial_balance)
                                3 LOAD_FAST                 0 (self)
                                6 STORE_ATTR                0 (balance)
```

```
                                9 LOAD_CONST                0 (None)
                               12 RETURN_VALUE

            2 (code object)
               co_argcount:        2
               co_cellvars:        ()
               co_filename:        './12.py'
               co_firstlineno:     4
               co_flags:           0x00043
               co_freevars:        ()
               co_lnotab:          '\x00\x01'
               co_name:            'deposit'
               co_names:           ('balance',)
               co_nlocals:         2
               co_stacksize:       3
               co_varnames:        ('self', 'amount')
               co_consts
                  0 None
               co_code
                  7c 00 00 04 6a 00 00 7c 01 00 37 02 5f 00 00 64
                  00 00 53
                  disassembled:
                    5           0 LOAD_FAST                 0 (self)
                                3 DUP_TOP
                                4 LOAD_ATTR                 0 (balance)
                                7 LOAD_FAST                 1 (amount)
                               10 INPLACE_ADD
                               11 ROT_TWO
                               12 STORE_ATTR                0 (balance)
                               15 LOAD_CONST                0 (None)
                               18 RETURN_VALUE

            3 (code object)
               co_argcount:        2
               co_cellvars:        ()
               co_filename:        './12.py'
               co_firstlineno:     6
               co_flags:           0x00043
               co_freevars:        ()
               co_lnotab:          '\x00\x01'
               co_name:            'withdraw'
               co_names:           ('balance',)
               co_nlocals:         2
               co_stacksize:       3
               co_varnames:        ('self', 'amount')
               co_consts
                  0 None
               co_code
                  7c 00 00 04 6a 00 00 7c 01 00 38 02 5f 00 00 64
                  00 00 53
                  disassembled:
                    7           0 LOAD_FAST                 0 (self)
                                3 DUP_TOP
                                4 LOAD_ATTR                 0 (balance)
                                7 LOAD_FAST                 1 (amount)
                               10 INPLACE_SUBTRACT
                               11 ROT_TWO
                               12 STORE_ATTR               0 (balance)
                               15 LOAD_CONST               0 (None)
                               18 RETURN_VALUE

            4 (code object)
               co_argcount:        1
               co_cellvars:        ()
               co_filename:        './12.py'
               co_firstlineno:     8
               co_flags:           0x00043
               co_freevars:        ()
               co_lnotab:          '\x00\x01'
               co_name:            'overdrawn'
               co_names:           ('balance',)
               co_nlocals:         1
               co_stacksize:       2
               co_varnames:        ('self',)
               co_consts
                  0 None
                  1 0
               co_code
                  7c 00 00 6a 00 00 64 01 00 6b 00 00 53
                  disassembled:
                    9            0 LOAD_FAST                0 (self)
                                 3 LOAD_ATTR                0 (balance)
                                 6 LOAD_CONST               1 (0)
                                 9 COMPARE_OP               0 (<)
                                12 RETURN_VALUE

         co_code
            65 00 00 5a 01 00 64 00 00 64 01 00 84 01 00 5a
            02 00 64 02 00 84 00 00 5a 03 00 64 03 00 84 00
            00 5a 04 00 64 04 00 84 00 00 5a 05 00 52 53
            disassembled:
              1           0 LOAD_NAME                0 (__name__)
                          3 STORE_NAME               1 (__module__)

              2           6 LOAD_CONST               0 (0)
                          9 LOAD_CONST               1 (<code object __init__ at
0x1048c16b0, file "./12.py", line 2>)
                         12 MAKE_FUNCTION            1
                         15 STORE_NAME               2 (__init__)

              4          18 LOAD_CONST               2 (<code object deposit at
0x1048c1630, file "./12.py", line 4>)
                         21 MAKE_FUNCTION            0
                         24 STORE_NAME               3 (deposit)

              6          27 LOAD_CONST               3 (<code object withdraw at
0x1048c1330, file "./12.py", line 6>)
                         30 MAKE_FUNCTION            0
                         33 STORE_NAME               4 (withdraw)

              8          36 LOAD_CONST               4 (<code object overdrawn
at 0x1048c12b0, file "./12.py", line 8>)
```

```
                         39 MAKE_FUNCTION               0
                         42 STORE_NAME                  5 (overdrawn)
                         45 LOAD_LOCALS
                         46 RETURN_VALUE

      2 15
      3 5
      4 None
   co_code
      64 00 00 65 00 00 66 01 00 64 01 00 84 00 00 83
      00 00 59 5a 01 00 65 01 00 64 02 00 83 01 00 5a
      02 00 65 02 00 6a 03 00 64 03 00 83 01 00 01 65
      02 00 6a 04 00 47 48 64 04 00 53
      disassembled:
        1           0 LOAD_CONST               0 ('BankAccount')
                    3 LOAD_NAME                0 (object)
                    6 BUILD_TUPLE              1
                    9 LOAD_CONST               1 (<code object BankAccount at
0x1048c17b0, file "./12.py", line 1>)
                   12 MAKE_FUNCTION            0
                   15 CALL_FUNCTION            0
                   18 BUILD_CLASS
                   19 STORE_NAME               1 (BankAccount)

       10          22 LOAD_NAME                1 (BankAccount)
                   25 LOAD_CONST               2 (15)
                   28 CALL_FUNCTION            1
                   31 STORE_NAME               2 (my_account)

       11          34 LOAD_NAME                2 (my_account)
                   37 LOAD_ATTR                3 (withdraw)
                   40 LOAD_CONST               3 (5)
                   43 CALL_FUNCTION            1
                   46 POP_TOP

       12          47 LOAD_NAME                2 (my_account)
                   50 LOAD_ATTR                4 (balance)
                   53 PRINT_ITEM
                   54 PRINT_NEWLINE
                   55 LOAD_CONST               4 (None)
                   58 RETURN_VALUE
```