2110316 Prog Lang Prin

Part A : Quiz (20 marks (15%) 40 minutes)

1. From the web pages https://www.quora.com/Is-Python-compiled-or-interpreted-or-both and
    http://stackoverflow.com/questions/6889747/is-python-interpreted-or-compiled-or-both

    Python is a dynamic language, and when running it from the command line, Python reads the .py into memory, and compiles it in order to get a bytecode, then goes on to execute. For example, the statement a = b.c() is compiled to a byte stream which, when "disassembled", looks somewhat like load 0 (b); load_str 'c'; get_attr; call_function 0; store 1 (a).

    For each module that is imported by the program, Python first checks to see whether there is a precompiled bytecode version, in a .pyo or .pyc, that has a timestamp which corresponds to its .py file. Python uses the bytecode version if there is. Otherwise, it compiles the imported module's .py file, saves it into a .pyc file, and uses the bytecode it just created.

    Once our program has been compiled into byte code, it is shipped off for execution to Python Virtual Machine (PVM). Actually, the PVM is just a big loop that iterates through our byte code instructions, one by one, to carry out their operations. Technically it's just the last step of what is called the Python interpreter.

1.1 (1 mark) Given the description above, how does the program translation in Python differ from translation using a traditional compiler?

Traditional compiler compiles source code into executable machine-specific code but Python compilation above compiles source code into Python-specific bytecode which cannot be executed on a machine yet (hence it needs to be interpreted further in a PVM.)

1.2 (2 marks) Give two opinions about why the implementation of Python performs program translation as above, rather than translating like a traditional compiler.

1. Python is a dynamic language and many details (e.g. type) cannot be determined at compile time, but rather at run time. Hence for language translation, it is easier to implement an interpreter (or a combination of compiler and interpreter as above) than a traditional compiler.

2. Compilation into machine-independent bytecode makes the program more portable across machines, provided that the machines have a PVM.

2. For the code below, the language uses a value model of variables.

```
program A()
{
    x, y, z: integer;

    procedure B()
    {
        y: integer;
        y=0;
        x=z+1;
        z=y+2;
    }

    procedure C()
    {
        z: integer;

        procedure D()
        {
            x: integer;
            x = z + 1;
            y = x + 1;
            call B();
        }

        z = 5;
        call D();
    }

    x = 10;
    y = 11;
    z = 12;
    call C();
    print x, y, z;
}
```

2.1 (3 marks) If the language uses static scoping, the printed result of x, y, and z is	2.2 (3 marks) If the language uses dynamic scoping, the printed result of x, y, and z is
x = 13	x = 10
y = 7	y = 7
z = 2	z = 12

3. The following information about Java is used for the questions 3.1-3.3.

Java type promotion rules:

Type promotion is an automatic type conversion from a "lesser" base type to a "greater" one. When an operator applies binary numeric promotion to a pair of operands, the following rules apply, in order, using widening conversion to convert operands as necessary:

If either operand is of type double, the other is converted to double.

Otherwise, if either operand is of type float, the other is converted to float.

Otherwise, if either operand is of type long, the other is converted to long.

Otherwise, both operands are converted to type int.

Java operator predecence and associativity:

Operator	Precedence Level (เรียงจาก level สูงลงไปต่ำ)	Associativity
...	...	...
( ) (cast) ...	3	Right to left
...	...	...
+ (additive) ...	5	Left to right
...	...	...
= (assignment) ...	15	Right to left

3.1 (2 marks)

```
int i = 5;                //line1
long j = 8;               //line2
i = i+j;                  //line3, compile error
i += j;                   //line4, compile OK
```

Why is there a compile error at line3?

i is promoted to long before being added to j. The result of i+j is a long. But Java compiler does not allow narrowing conversion from long to int because of possible loss of precision.

3.2 (1 mark) Why does line4 above, which is generally known to be equivalent to line3, not get a compile error? (Hint: How can you fix line3?)

The operator += in line4 works like i = (int)(i+j). (That is how we can fix line3.)

3.3 (2 marks)

```
byte a = 1;                //line5
int b = 2;                 //line6
byte c = (byte)  a+b;      //line7, compile error
```

Why is there a compile error when casting the addition between a and b at line7?

Cast has higher precedence than +, and line7 works as byte c = ((byte)a) + b. So (byte)a is promoted to int before adding to b. The result is int so it cannot be assigned to c as narrowing conversion from int to byte is not allowed.

4. Given the C++ code below, choose an answer and fill in the blank for each question.

```
class Mammal {
public:
    void giveBirth() { cout << "I give birth to live young." << endl; }
    virtual void live() { cout << "I live everywhere." << endl; }
};
class Platypus : public Mammal {          //subclass of Mammal
public:
    void giveBirth() { cout << "I lay eggs." << endl; }
    virtual void live() { cout << "I live in Australia." << endl; }
};
class PlatypusChild : public Platypus {       //subclass of Platypus
public:
    void giveBirth() { cout << "I lay eggs too." << endl; }
    virtual void live() { cout << "I live in Australia too." << endl; }
};

void fnGiveBirth(Mammal *mm) { mm->giveBirth(); }
void fnLive(Mammal *mm) { mm->live(); }

int main ()
{
    Mammal *aMammal = new Platypus();
    PlatypusChild aPlatypusChild;
    Platypus aPlatypus = aPlatypusChild;
    fnLive(aMammal);                  //line1
    fnGiveBirth(&aPlatypus);          //line2
    fnLive(&aPlatypusChild);          //line3
}
```

4.1 (2 marks) Which parameter passing mode is used at line1?

☑ call by value ☐ call by reference ☐ call by sharing

What is the printed result of line1? ...I live in Australia..

4.2 (2 marks) Which method binding is used at line2?

☐ dynamic method binding ☑ static method binding

What is the printed result of line2? ...I give birth to live young..

4.3 (2 marks) Which method binding is used at line3?

☑ dynamic method binding ☐ static method binding

What is the printed result of line3? ...I live in Australia too. ..