


<b>Giảng viên ra đề:</b> (Chữ ký và Họ tên)	(Ngày ra đề)	<b>Người phê duyệt:</b> (Chữ ký và họ tên)	(Ngày duyệt đề)

 TRƯỜNG ĐH BÁCH KHOA - ĐHQG-HCM <b><u>KHOA KH &amp; KT MÁY TÍNH</u></b>	<b>THI CUỐI KỲ</b>		Học kỳ / Năm học		1	2020-2021
			Ngày thi		16-01-2021	
	Môn học	Nguyên lý ngôn ngữ lập trình				
	Mã môn học	CO3005				
	Thời lượng	120 phút	Mã đề	2010		
<b>Ghi chú:</b> - Sinh viên không được phép sử dụng tài liệu. - Sinh viên nộp lại đề sau khi kiểm tra.						

## I. PHẦN LẬP TRÌNH (đồng thời dùng để tính điểm B của các bài tập lớn 3 và 4) (4 điểm)

**Câu 1.** [L.O.3.1] (2 điểm) Cho các lớp của AST được khai báo vắn tắt như sau:

```
class AST(ABC)
class Decl(AST)
class Type(AST)
class TypeDecl(Decl): #name:str, rhs:Type
class VarDecl(Decl): # name:str, rhs:Type
class Block(Decl): #ele:list(Decl)
class IntType(Type)
class FloatType(Type)
class Id(Type): #name:str
```

Ví dụ:

```
Block([VarDecl("a", IntType()),
      TypeDecl("vd", FloatType()),
      Block([VarDecl("b", FloatType()),
            VarDecl("c", Id("vd"))]),
      Block([VarDecl("d", IntType()),
            TypeDecl("vd1", IntType())]),
      VarDecl("e", Id("vd"))])
```

Block 1: b:6, c:6

Block 2: d: 2

Block: a:2, max(Block1,Block2) = 12, e:6 = 20

a=2,padding=10,Block2:12,e:6,padding:6

12 24 36 48

là một AST hợp lệ.

Hãy viết một visitor StaticCheck để thực hiện các yêu cầu sau:

- Kiểm tra một tên kiểu được khai báo chưa:** Một tên kiểu chỉ được sử dụng sau khi nó được khai báo bởi một TypeDecl trong cùng Block hoặc ở một Block bao bên ngoài. Nếu không thỏa điều kiện này thì báo lỗi bằng cách ném biến cố UndeclaredType(name:str). Trong ví dụ trên, khai báo biến c dùng tên kiểu vd hợp lệ vì vd được khai báo trước ở Block bao bên ngoài; khai báo e dùng tên kiểu vd cũng hợp lệ vì vd được khai báo trước trong cùng Block. Nếu biến e được khai báo tên kiểu vd1 thì sẽ gây ra lỗi UndeclaredType(vd1) vì vd1 được khai báo trước nhưng thuộc Block bên trong.
- Tính toán kích thước của Block:** Kích thước của Block gồm tổng kích thước của các kiểu trong các khai báo biến (VarDecl) và kích thước của Block có kích thước lớn nhất trong Block. Cho biết kích thước của IntType() là 2, FloatType() là 6 và kích thước của Id là kích thước của kiểu tương ứng. Trong ví dụ trên kích thước của Block ngoài cùng là 20 (với 2 từ biến a, 12 từ Block thứ nhất, 6 từ biến e).

Các kiểu + Block lớn nhất

IntType: 2

FloatType: 6

Id:

**Câu 2.** [L.O.3.1] (2 điểm) Viết phương thức visitWhile(self,ast:While,o:Object) cho lớp CodeGeneration để sinh mã cho phát biểu While. Cho biết node While trên AST được khai báo bởi class sau:

```
class While(Stmt): #exp:Expr, body:list(Stmt), else:list(Stmt)
```

Ngữ nghĩa của phát biểu while trên tương tự ngữ nghĩa của phát biểu while trên Python. Chú ý đặt các nhãn ứng với vị trí đích của các phát biểu continue và break bên trong body.

Các phương thức của Emitter có thể được dùng:

- emitIFTRUE(self, label: int, frame)
- emitIFFALSE(self, label: int, frame)
- emitGOTO(self, label: int, frame)
- emitLABEL(self, label: int, frame)
- emitREADVAR(self, name, intype, index, frame)
- emitWRITEVAR(self, name, intype, index, frame)
- emitPUSHICONST(self, in: int, frame)
- emitADDOP(self, op: str, intype, frame)
- emitREOP(self, op: str, intype, frame)

- prinout(self, out: str)

Các phương thức của Frame có thể được dùng:

- enterLoop(self)
- exitLoop(self)
- enterScope(self)
- exitScope(self)
- getNewIndex(self)
- getNewLabel(self)
- getBreakLabel(self)
- getContinueLabel(self)

## II. PHẦN LÝ THUYẾT (6 điểm)

**Câu 3.** [L.O.2.1] (2 điểm)

Trong một ngôn ngữ có cấu trúc khối tựa C/C++, cho một đoạn mã sau đây:

<pre>1 int z = 0; 2 int foo() { return foo(); } 3 int bar(int x, int y) { 4     double inbar(double z) { return 5         if (x == 0) return x; else return x + y; 6     } 7 cout &lt;&lt; foo(z, foo() + z);</pre>	<p>1. Pass by name:</p> <pre>x==z, y==foo()+z inbar(z) -&gt; z+z if z==0 -&gt; z else -&gt; z+foo()+z Output: 0</pre>	<p>2. Pass by value:</p> <pre>bar(x=0,y=k), foo()=k Output: 0</pre>
---	---	---

Biết rằng, **cout** là lệnh dùng để xuất ra màn hình. Hãy cho biết **giải thích** kết quả thực thi của đoạn mã trên nếu:

- Tham số được truyền cho hàm bar theo tên (pass by name).
- Tham số được truyền cho hàm bar theo giá trị (pass by value).
- Hãy cho biết môi trường tham khảo của hàm inbar được khai báo ở dòng số 4.

**Câu 4.** [L.O.2.1] (2 điểm)

Hãy suy diễn kiểu cho các hàm E, F và G cho một đoạn trong một chương trình được viết bằng ngôn ngữ X sau:

```
// Definition of U, Z
def F(t, x, y) = if E(x) then 0 else t(x, y)
def G(m, n) = n(m)
F(G, z, U)
```

F:  $T1 * T2 * T3 \rightarrow T4$       E(x) --> Type of x = Input type of E -->  $T2 = T5$   
 t:  $T1$ , x:  $T2$ , y:  $T3$       t is a function with 2 paras:  $T1 = T7 * T8 \rightarrow T9$   
 E:  $T5 \rightarrow T6$       E(x) is Boolean expression -->  $T6 = \text{Boolean}$   
 F returns 0 or t(x,y) --> Output type of t is Int (0) -->  $T9 = \text{Int}$   
 x,y are 2 paras of t -->  $T7 = T2, T8 = T3$  Conclusion:  $F: (T2 * T3 \rightarrow \text{Int}) * T2 * T3 \rightarrow \text{Int}$   
 E:  $T2 \rightarrow \text{Boolean}$

Biết rằng, các hàm trong ngôn ngữ X không thể quá tải hàm (overload) hay danh sách kiểu của tham số đầu vào và kiểu trả về của hàm là duy nhất. Biểu thức điều kiện phải có kiểu **boolean**, các giá trị trả về trên các nhánh thực thi của chương trình phải có cùng kiểu. Hằng 0 là giá trị có kiểu **integer**. Giả sử kiểu của z và U là T1 và T2. Yêu cầu, kết quả suy diễn kiểu sử dụng ít biến kiểu nhất có thể.

**Câu 5.** [L.O.2.1] (2 điểm)      G:  $(T2 * T3) * (T2 * T3 \rightarrow \text{Int}) \rightarrow \text{Int}$       n:  $T10 \rightarrow T11$        $T10 == T2 * T3, T11 == \text{Int}$

(a) Trình cộng hành (coroutine) là gì? Điểm khác biệt giữa trình cộng hành và cơ chế gọi-trở về đơn giản là gì? Hãy nêu một ứng dụng của trình cộng hành.

(b) Làm thế nào hiện thực một trình cộng hành trên Python3? Hãy viết hai chương trình con chạy theo chế độ trình cộng hành sao cho hai trình luân phiên nhau in các giá trị 1, 2, 3, 4, 5, 6 (một trình in 1, 3, 5 và trình còn lại in 2, 4, 6).

Coroutine	Call-return	def func():	G==t,t:
A coroutine may postpone its execution and control is back to caller	No recursive	yield 1	$T2 * T3 \rightarrow \text{Int}$
Its execution later is resumed at the place it postponed	Explicit Call Site	yield 3	$T2 * T3 == (T2 * T3) * (T2 * T3 \rightarrow \text{Int})$
	Single Entry Point	yield 5	Trang 2/2- Mã đề thi 2010
	Immediate Control Passing		
	Single Execution		