# [CC03] AST Programming Code

### Question 1

Correct

Marked out of 1.00

Flag question

#### Question text

Given the grammar of MP as follows:

program: vardecls EOF;

vardecls: vardecl vardecltail;

vardecltail: vardecl vardecltail | ;

vardecl: mptype ids ';' ;

mptype: INTTYPE | FLOATTYPE;

ids: ID ',' ids | ID;

INTTYPE: 'int';

FLOATTYPE: 'float';

ID: [a-z]+ ;

Please copy the following class into your answer and modify the bodies of its methods to count the terminal nodes in the parse tree? Your code starts at line 10.

class TerminalCount(MPVisitor):

    def visitProgram(self,ctx:MPParser.ProgramContext):

        return None

    def visitVardecls(self,ctx:MPParser.VardeclsContext):

        return None

    def visitVardecltail(self,ctx:MPParser.VardecltailContext):

        return None

    def visitVardecl(self,ctx:MPParser.VardeclContext):

        return None

    def visitMptype(self,ctx:MPParser.MptypeContext):

        return None

    def visitIds(self,ctx:MPParser.IdsContext):

        return None

Answer:(penalty regime: 0 %)

1

20

class TerminalCount(MPVisitor):

def visitProgram(self, ctx:MPParser.ProgramContext):

return 1 + self.visit(ctx.vardecls())

def visitVardecls(self, ctx:MPParser.VardeclsContext):

return self.visit(ctx.vardecl()) + self.visit(ctx.vardecltail())

def visitVardecltail(self, ctx:MPParser.VardecltailContext):

return self.visit(ctx.vardecl()) + self.visit(ctx.vardecltail()) if ctx.vardecl() else 0

def visitVardecl(self, ctx:MPParser.VardeclContext):

return 1 + self.visit(ctx.mptype()) + self.visit(ctx.ids())

def visitMptype(self, ctx:MPParser.MptypeContext):

return 1

def visitIds(self, ctx:MPParser.IdsContext):

return 2 + self.visit(ctx.ids()) if ctx.ids() else 1

CheckQuestion 1

#### Feedback

|  | **Test** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | "int a;" | 4 | 4 |  |
|  | """int a,b;""" | 6 | 6 |  |
|  | "int a;float b;" | 7 | 7 |  |
|  | "int a,b;float c;" | 9 | 9 |  |
|  | "int a,b;float c,d,e;" | 13 | 13 |  |

Passed all tests!

### Question 2

Correct

Marked out of 1.00

Flag question

#### Question text

Given the grammar of MP as follows:

program: vardecls EOF;

vardecls: vardecl vardecltail;

vardecltail: vardecl vardecltail | ;

vardecl: mptype ids ';' ;

mptype: INTTYPE | FLOATTYPE;

ids: ID ',' ids | ID;

INTTYPE: 'int';

FLOATTYPE: 'float';

ID: [a-z]+ ;

Please copy the following class into your answer and modify the bodies of its methods to return the height of the parse tree? Your code starts at line 10.

class Height(MPVisitor):

    def visitProgram(self,ctx:MPParser.ProgramContext):

        return None

    def visitVardecls(self,ctx:MPParser.VardeclsContext):

        return None

    def visitVardecltail(self,ctx:MPParser.VardecltailContext):

        return None

    def visitVardecl(self,ctx:MPParser.VardeclContext):

        return None

    def visitMptype(self,ctx:MPParser.MptypeContext):

        return None

    def visitIds(self,ctx:MPParser.IdsContext):

        return None

Answer:(penalty regime: 0 %)

1

20

class Height(MPVisitor):

def visitProgram(self,ctx:MPParser.ProgramContext):

return 1 + self.visit(ctx.vardecls())

def visitVardecls(self,ctx:MPParser.VardeclsContext):

return 1 + max(self.visit(ctx.vardecl()),self.visit(ctx.vardecltail()))

def visitVardecltail(self,ctx:MPParser.VardecltailContext):

if ctx.vardecl(): return 1 + max(self.visit(ctx.vardecl()),self.visit(ctx.vardecltail()))

else: return 0

def visitVardecl(self,ctx:MPParser.VardeclContext):

return 1+ max(self.visit(ctx.mptype()),self.visit(ctx.ids()))

def visitMptype(self,ctx:MPParser.MptypeContext):

return 2

def visitIds(self,ctx:MPParser.IdsContext):

if ctx.ids(): return 1+self.visit(ctx.ids())

else: return 2

CheckQuestion 2

#### Feedback

|  | **Test** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | "int a;" | 5 | 5 |  |
|  | """int a,b;""" | 6 | 6 |  |
|  | "int a;float b;" | 6 | 6 |  |
|  | "int a,b;float c;" | 6 | 6 |  |
|  | "int a,b;float c,d,e;" | 8 | 8 |  |

Passed all tests!

Top of Form

### Question 3

Correct

Marked out of 1.00

Flag question

#### Question text

Given the grammar of MP as follows:

program: vardecls EOF;

vardecls: vardecl vardecltail;

vardecltail: vardecl vardecltail | ;

vardecl: mptype ids ';' ;

mptype: INTTYPE | FLOATTYPE;

ids: ID ',' ids | ID;

INTTYPE: 'int';

FLOATTYPE: 'float';

ID: [a-z]+ ;

Please copy the following class into your answer and modify the bodies of its methods to count the internal nodes in the parse tree?

class NonTerminalCount(MPVisitor):

    def visitProgram(self,ctx:MPParser.ProgramContext):

        return None

    def visitVardecls(self,ctx:MPParser.VardeclsContext):

        return None

    def visitVardecltail(self,ctx:MPParser.VardecltailContext):

        return None

    def visitVardecl(self,ctx:MPParser.VardeclContext):

        return None

    def visitMptype(self,ctx:MPParser.MptypeContext):

        return None

    def visitIds(self,ctx:MPParser.IdsContext):

        return None

Answer:(penalty regime: 0 %)

1

30

class NonTerminalCount(MPVisitor):

def visitProgram(self, ctx:MPParser.ProgramContext):

return self.visit(ctx.vardecls()) + 1

def visitVardecls(self, ctx:MPParser.VardeclsContext):

count = 0

if ctx.vardecl() is not None:

count += 1 + self.visit(ctx.vardecl()) + self.visit(ctx.vardecltail())

return count

def visitVardecltail(self, ctx:MPParser.VardecltailContext):

count = 0

if ctx.vardecl() is not None:

count += self.visit(ctx.vardecl()) + self.visit(ctx.vardecltail())

return count

def visitVardecl(self, ctx:MPParser.VardeclContext):

return 3 + self.visit(ctx.ids())

def visitMptype(self, ctx:MPParser.MptypeContext):

return 1

def visitIds(self, ctx:MPParser.IdsContext):

count = 0

if ctx.ID() is not None:

count += 1

if ctx.ids() is not None:

count += self.visit(ctx.ids())

return count

CheckQuestion 3

#### Feedback

|  | **Test** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | "int a;" | 6 | 6 |  |
|  | """int a,b;""" | 7 | 7 |  |
|  | "int a;float b;" | 10 | 10 |  |
|  | "int a,b;float c;" | 11 | 11 |  |
|  | "int a,b;float c,d,e;" | 13 | 13 |  |

Passed all tests!

Bottom of Form

Top of Form

### Question 4

Correct

Marked out of 1.00

Flag question

#### Question text

Given the grammar of MP as follows:

program: vardecls EOF;

vardecls: vardecl vardecltail;

vardecltail: vardecl vardecltail | ;

vardecl: mptype ids ';' ;

mptype: INTTYPE | FLOATTYPE;

ids: ID ',' ids | ID;

INTTYPE: 'int';

FLOATTYPE: 'float';

ID: [a-z]+ ;

and AST classes as follows:

class Program:#decl:list(VarDecl)

class Type(ABC): pass

class IntType(Type): pass

class FloatType(Type): pass

class VarDecl: #variable:Id; varType: Type

class Id: #name:str

Please copy the following class into your answer and modify the bodies of its methods to generate the AST of a MP input?

class ASTGeneration(MPVisitor):

    def visitProgram(self,ctx:MPParser.ProgramContext):

        return None

    def visitVardecls(self,ctx:MPParser.VardeclsContext):

        return None

    def visitVardecltail(self,ctx:MPParser.VardecltailContext):

        return None

    def visitVardecl(self,ctx:MPParser.VardeclContext):

        return None

    def visitMptype(self,ctx:MPParser.MptypeContext):

        return None

    def visitIds(self,ctx:MPParser.IdsContext):

        return None

Answer:(penalty regime: 0 %)

1

36

class ASTGeneration(MPVisitor):

def visitProgram(self,ctx:MPParser.ProgramContext):

vardecls = self.visit(ctx.vardecls())

return Program(vardecls)

def visitVardecls(self,ctx:MPParser.VardeclsContext):

vardecl = self.visit(ctx.vardecl())

vardecltail = self.visit(ctx.vardecltail())

if vardecltail is None: return vardecl

return vardecl + vardecltail

def visitVardecltail(self,ctx:MPParser.VardecltailContext):

if ctx.vardecltail() is None:

return None

vardecl = self.visit(ctx.vardecl())

vardecltail = self.visit(ctx.vardecltail())

if vardecltail is None: return vardecl

return vardecl + vardecltail

def visitVardecl(self,ctx:MPParser.VardeclContext):

typ = self.visit(ctx.mptype())

ids = self.visit(ctx.ids())

list\_var = []

for i in ids:

list\_var.append(VarDecl(i,typ))

return list\_var

def visitMptype(self,ctx:MPParser.MptypeContext):

if ctx.INTTYPE():

return IntType()

return FloatType()

def visitIds(self,ctx:MPParser.IdsContext):

if ctx.ids() is None:

return [Id(ctx.ID().getText())]

return [Id(ctx.ID().getText())] + self.visit(ctx.ids())

CheckQuestion 4

#### Feedback

|  | **Test** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | "int a;" | Program([VarDecl(Id(a),IntType)]) | Program([VarDecl(Id(a),IntType)]) |  |
|  | """int a,b;""" | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType)]) | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType)]) |  |
|  | "int a;float b;" | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),FloatType)]) | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),FloatType)]) |  |
|  | "int a,b;float c;" | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType),VarDecl(Id(c),FloatType)]) | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType),VarDecl(Id(c),FloatType)]) |  |
|  | "int a,b;float c,d,e;" | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType),VarDecl(Id(c),FloatType),VarDecl(Id(d),FloatType),VarDecl(Id(e),FloatType)]) | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType),VarDecl(Id(c),FloatType),VarDecl(Id(d),FloatType),VarDecl(Id(e),FloatType)]) |  |

Passed all tests!

Bottom of Form

Top of Form

### Question 5

Correct

Marked out of 1.00

Flag question

#### Question text

Given the grammar of MP as follows:

program: vardecl+ EOF;

vardecl: mptype ids ';' ;

mptype: INTTYPE | FLOATTYPE;

ids: ID (',' ID)\*;

INTTYPE: 'int';

FLOATTYPE: 'float';

ID: [a-z]+ ;

and AST classes as follows:

class Program:#decl:list(VarDecl)

class Type(ABC): pass

class IntType(Type): pass

class FloatType(Type): pass

class VarDecl: #variable:Id; varType: Type

class Id: #name:str

Please copy the following class into your answer and modify the bodies of its methods to generate the AST of a MP input?

class ASTGeneration(MPVisitor):

    def visitProgram(self,ctx:MPParser.ProgramContext):

        return None

    def visitVardecl(self,ctx:MPParser.VardeclContext):

        return None

    def visitMptype(self,ctx:MPParser.MptypeContext):

        return None

    def visitIds(self,ctx:MPParser.IdsContext):

        return None

Answer:(penalty regime: 0 %)

1

30

class ASTGeneration(MPVisitor):

def visitProgram(self,ctx:MPParser.ProgramContext):

body = []

for decl in ctx.vardecl():

body = body + self.visit(decl)

return Program(body)

def visitVardecl(self,ctx:MPParser.VardeclContext):

typ = self.visit(ctx.mptype())

ids = self.visit(ctx.ids())

list\_var = []

for i in ids:

list\_var.append(VarDecl(i,typ))

return list\_var

def visitMptype(self,ctx:MPParser.MptypeContext):

if ctx.INTTYPE():

return IntType()

return FloatType()

def visitIds(self,ctx:MPParser.IdsContext):

list\_id=ctx.ID()

result = []

for i in list\_id:

result.append(Id(i.getText()))

return result

CheckQuestion 5

#### Feedback

|  | **Test** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | "int a;" | Program([VarDecl(Id(a),IntType)]) | Program([VarDecl(Id(a),IntType)]) |  |
|  | """int a,b;""" | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType)]) | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType)]) |  |
|  | "int a;float b;" | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),FloatType)]) | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),FloatType)]) |  |
|  | "int a,b;float c;" | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType),VarDecl(Id(c),FloatType)]) | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType),VarDecl(Id(c),FloatType)]) |  |
|  | "int a,b;float c,d,e;" | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType),VarDecl(Id(c),FloatType),VarDecl(Id(d),FloatType),VarDecl(Id(e),FloatType)]) | Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType),VarDecl(Id(c),FloatType),VarDecl(Id(d),FloatType),VarDecl(Id(e),FloatType)]) |  |

Passed all tests!

Bottom of Form