CSE 574

Project Phase 4 Executable code

Akhilesh Kumar, Yuanyuan Zhang

**Grammar**

|  |  |
| --- | --- |
| **Production** | **Body** |
| Program | Function\_Declaration **begin** Statement\_list **end** |
| Function\_Declaration | Method Function\_Declaration | Method |
| Method | **procedure** int Ident ( ) **begin** Statement\_list **end** | **procedure** float Ident ( ) **begin** Statement\_list **end**| **procedure** string Ident ( ) **begin** Statement\_list **end**| **procedure** **void** Ident ( ) **begin** Statement\_list **end** |
| Statement**\_**list | Statement | Statement Statement\_list |
|  |  |
| Statement | **ID** := Expror | **read** ( Id\_list ) | **write** ( Expression\_list ) |  **int** id\_list | **float** id\_list | **string** id\_list | **if (expr) then** Statement\_list **endif** **| if (**Expression**) then** Statement\_list **else** Statement\_list **endif | while (**Expror**) begin** Statement\_list **end** | Function\_Call | ID := Function\_Call | **return** ident | **return** |
| Function\_Call | Ident( ) |
| Id\_list | Ident | Ident, Id\_list |
| Expression\_list | Expror | Expression\_list**,** Expror |
| Expror | Expand | Expror || Expand |
| Expand | Expeq | Expand && Expeq |
| Exeeq | Explogical | Expeq == Explogical | Expeq != Explogical |
| Expr\_logical | Expr | Exp\_logical > Expr | Exp\_logical < Expr | Exp\_logical >= Expr | Exp\_logical <= Expr |
| Expr | Term | Expr + Term | Expr - Term |
| Term | Factor | Term \* Factor | Term / Factor | Term % Factor |
| Factor | Primary | ! Primary |
| Primary | ( Expression ) | **ID** | **INTLITERAL | FLOATLITERAL | STRINGLITERAL** |
| Ident | **ID** |
| System\_goal | Program **SCANEOF** |

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Program** | **Expected Output** | **Actual Output** |
| 1. int (with return value)   **procedure** int foo ()  **begin**  int a,b,c;  a := b+c;  write(a);  return a;  **end**  **begin**  int c,d,e;  c := d+e;  e := foo();  **end** | procedure int foo()  a =0  b =0  c =0  t1=b+c  a=t1  write a  return a  c =0  d =0  e =0  t2 =d+e  c =t2  e = Call foo(), 0 | **same as expected**  Line 1: procedure int foo ()  Line 2: a = 0  Line 3: b = 0  Line 4: c = 0  Line 5: t1 = b + c  Line 6: a = t1  Line 7: write a  Line 8: return a  Line 9: c = 0  Line 10: d = 0  Line 11: e = 0  Line 12: t2 = d + e  Line 13: c = t2  Line 14: e = Call foo(), 0 |
| 1. float (with return value)   **procedure** float foo ()  **begin**  float a,b,c;  a := b+c;  write(a);  return a;  **end**  **begin**  float c,d,e;  c := d+e;  e := foo();  **end** | procedure float foo()  a =0.0  b =0.0  c =0.0  t1=b+c  a=t1  write a  return a  c =0.0  d =0.0  e =0.0  t2 =d+e  c =t2  e = Call foo(), 0 | **same as expected**  Line 1: procedure float foo ()  Line 2: a = 0.0  Line 3: b = 0.0  Line 4: c = 0.0  Line 5: t1 = b + c  Line 6: a = t1  Line 7: write a  Line 8: return a  Line 9: c = 0.0  Line 10: d = 0.0  Line 11: e = 0.0  Line 12: t2 = d + e  Line 13: c = t2  Line 14: e = Call foo(), 0 |
| (3) string (with return value)  **procedure** string foo ()  **begin**  string a,b;  a := b;  write(a);  return a;  **end**  **begin**  int c,d,e;  c := d+e;  string g;  g := foo();  **end** | procedure string foo()  string a  string b  a = b  write a  return a  c =0  d =0  e =0  t1 =d+e  c =t1  string g  f = Call foo(), 0 | **same as expected**  Line 1: procedure string foo ()  Line 2: string a  Line 3: string b  Line 4: a = b  Line 5: write a  Line 6: return a  Line 7: c = 0  Line 8: d = 0  Line 9: e = 0  Line 10: t1 = d + e  Line 11: c = t1  Line 12: string g  Line 13: f = Call foo(), 0 |
| 1. multiple procedures using same variable name(a used twice in 2 procedures)   **procedure** int foo ()  begin  int a,b,c;  a := b+c;  write(a);  return b;  **end**  **procedure** int bar ()  **begin**  int a, c, d, e;  a:=d+e;  d:=foo();  write(c);  return c;  **end**  **begin**  int f, g, h;  f := g + h;  h := bar();  **end** | procedure int foo()  a =0  b =0  c =0  t1=b+c  a =t1  write a  return b  procedure int bar()  a = 0  c = 0  d = 0  e = 0  t2=d+e  a = t2  d = call foo(), 0  write c  return c  f =0  g =0  h =0  t3 = g + h  f = t3  h = call bar() | **same as expected**  Line 1: procedure int foo ()  Line 2: a = 0  Line 3: b = 0  Line 4: c = 0  Line 5: t1 = b + c  Line 6: a = t1  Line 7: write a  Line 8: return b  Line 9: procedure int bar ()  Line 10: a = 0  Line 11: c = 0  Line 12: d = 0  Line 13: e = 0  Line 14: t2 = d + e  Line 15: a = t2  Line 16: d = Call foo(),0  Line 17: write c  Line 18: return c  Line 19: f = 0  Line 20: g = 0  Line 21: h = 0  Line 22: t3 = g + h  Line 23: f = t3  Line 24: h = Call bar(),0 |
| 1. multiple procedures for int   **procedure** int foo ()  **begin**  int a,b,c;  a := b+c;  write(a);  return a;  **end**  **procedure** int bar ()  **begin**  int c, d, e;  d:=foo();  write(c);  return c;  **end**  **begin**  int f, g, h;  f := g + h;  h := bar();  **end** | procedure int foo()  a =0  b =0  c =0  t1=b+c  a = t1  write a  return a  procedure int bar()  c =0  d =0  e =0  d = call foo(), 0  write c  return c  f =0  g =0  h =0  t2= g + h  f =t2  h = call bar(), 0 | **same as expected**  Line 1: procedure int foo ()  Line 2: a = 0  Line 3: b = 0  Line 4: c = 0  Line 5: t1 = b + c  Line 6: a = t1  Line 7: write a  Line 8: return a  Line 9: procedure int bar ()  Line 10: c = 0  Line 11: d = 0  Line 12: e = 0  Line 13: d = Call foo(), 0  Line 14: write c  Line 15: return c  Line 16: f = 0  Line 17: g = 0  Line 18: h = 0  Line 19: t2 = g + h  Line 20: f = t2  Line 21: h = Call bar(), 0 |
| 1. multiple procedures   **procedure** float foo ()  **begin**  float a,b,c;  a := b+c;  write(a);  return a;  **end**  **procedure** int bar ()  **begin**  float c, d, e;  foo();  write(c);  return c;  **end**  **begin**  float f, g, h;  f := g + h;  h := bar();  **end** | procedure float foo()  a =0.0  b =0.0  c =0.0  t1 =b+c  a =t1  write a  return a  procedure int bar()  c =0.0  d =0.0  e =0.0  call foo() ,0  write c  return c  f =0.0  g =0.0  h =0.0  t2= g + h  f =t2  h = call bar(), 0 | **same as expected**  Line 1: procedure float foo ()  Line 2: a = 0.0  Line 3: b = 0.0  Line 4: c = 0.0  Line 5: t1 = b + c  Line 6: a = t1  Line 7: write a  Line 8: return a  Line 9: procedure int bar ()  Line 10: c = 0.0  Line 11: d = 0.0  Line 12: e = 0.0  Line 13: Call foo(), 0  Line 14: write c  Line 15: return c  Line 16: f = 0.0  Line 17: g = 0.0  Line 18: h = 0.0  Line 19: t2 = g + h  Line 20: f = t2  Line 21: h = Call bar(), 0 |
| (7) multiple procedures for string  **procedure** string foo ()  **begin**  string a,b;  b := “cd”  a := b;  write(a);  return a;  **end**  **procedure** int bar ()  **begin**  int c,d, e;  foo();  write(c);  return c;  **end**  **begin**  string f, g;  g = bar();  **end** | procedure string foo()  string a  string b  b = “cd”  a = b  write a  return a  procedure int bar()  c =0  d =0  e =0  call foo() , 0  write c  return c  string g  g = call bar(), 0 | **same as expected**  Line 1: procedure string foo ()  Line 2: string a  Line 3: string b  Line 4: b = “cd”  Line 5: a = b  Line 6: write a  Line 7: return a  Line 8: procedure int bar ()  Line 9: c = 0  Line 10: d = 0  Line 11: e = 0  Line 12: Call foo(), 0  Line 13: write c  Line 14: return c  Line 15: string g  Line 16: g = Call bar(), 0 |
| 1. int , error when variable is not declared   **procedure** int foo ()  **begin**  int a,b;  a := b+c;  write(a);  return a;  **end**  **begin**  int c, d,e;  c := d+e;  e := foo();  **end** | This should return an error since there is no declaration for c in procedure foo, although it is defined later. | same as expected  12: error: Undefined identifier c is invalid rvalue |
| 1. int (with wrong return value)   **procedure** int foo ()  **begin**  int a,b;  string c;  a := b+c;  c:=”cd”;  write(a);  return c;  **end**  **begin**  int c,d,e;  c := d+e;  e := foo();  **end** | error: Invalid return type. | **same as expected**  14: error: return type do not match |
| 1. float (with wrong return value)   **procedure** float foo ()  **begin**  float a,b;  string c;  a := b+c;  c:=”cd”;  write(a);  return c;  **end**  **begin**  int c,d,e;  c := d+e;  e := foo();  **end** | error: invalid return type | **same as expected**  14: error: return type do not match |
| 1. string (with wrong return value)   **procedure** string foo ()  **begin**  int a,b;  string c;  a := b+c;  c:=”cd”;  write(a);  return a;  **end**  **begin**  int c,d,e;  c := d+e;  e := foo();  **end** | error: invalid return type | **same as expected**  14: error: return type do not match |
| 1. void procedure   **procedure** void foo ()  **begin**  int a,b,c;  a := b+c;  write(a);  return;  **end**  **begin**  int c,d,e;  c := d+e;  foo();  **end** | procedure void foo()  a =0  b =0  c =0  t1=b+c  a =t1  write a  return  c =0  d =0  e =0  t2 =d+e  c =t2  call foo(),0 | **same as expected**  Line 1: procedure void foo ()  Line 2: a = 0  Line 3: b = 0  Line 4: c = 0  Line 5: t1 = b + c  Line 6: a = t1  Line 7: write a  Line 8: return  Line 9: c = 0  Line 10: d = 0  Line 11: e = 0  Line 12: t2 = d + e  Line 13: c = t2  Line 14: Call foo(), 0 |
| 1. multiple procedures (with void)   **procedure** void foo ()  **begin**  float a,b,c;  a := b+c;  write(a);  return ;  **end**  **procedure** int bar ()  **begin**  float c, d, e;  foo();  write(c);  return c;  **end**  **begin**  int f, g, h;  f := g + h;  h := bar();  **end** | procedure void foo()  a =0.0  b =0.0  c =0.0  t1 =b+c  a = t1  write a  return  procedure int bar()  c = 0.0  d = 0.0  e = 0.0  call foo() ,0  write c  return c  f =0  g =0  h =0  t2 = g + h  f =t2  h = call bar(), 0 | **same as expected**  Line 1: procedure void foo ()  Line 2: a = 0.0  Line 3: b = 0.0  Line 4: c = 0.0  Line 5: t1 = b + c  Line 6: a = t1  Line 7: write a  Line 8: return  Line 9: procedure int bar ()  Line 10: c = 0.0  Line 11: d = 0.0  Line 12: e = 0.0  Line 13: Call foo(), 0  Line 14: write c  Line 15: return c  Line 16: f = 0  Line 17: g = 0  Line 18: h = 0  Line 19: t2 = g + h  Line 20: f = t2  Line 21: h = Call bar(), 0 |
| 1. multiple procedures   **procedure** int foo ()  **begin**  int a,b,c;  a := b+c;  write(a);  return a;  **end**  **procedure** int bar ()  **begin**  int c, d, e;  d:=foo();  a:=d;  write(c);  return c;  **end**  **begin**  int f, g, h;  f := g + h;  h := bar();  **end** | error : identifier ‘a’ not defined in procedure bar. | **same as expected**  20: error: Undefined identifier a is invalid variable name in the procedure 2 |
| 1. Invalid function call   **procedure** int foo ()  **begin**  int a,b,c;  a := b+c;  write(a);  return a;  **end**  **begin**  int f, g, h;  f := g + h;  h := bar();  **end** | invalid procedure name bar | **same as expected**  12: error: invalid procedure name bar |