Code generation

What is Code generation?

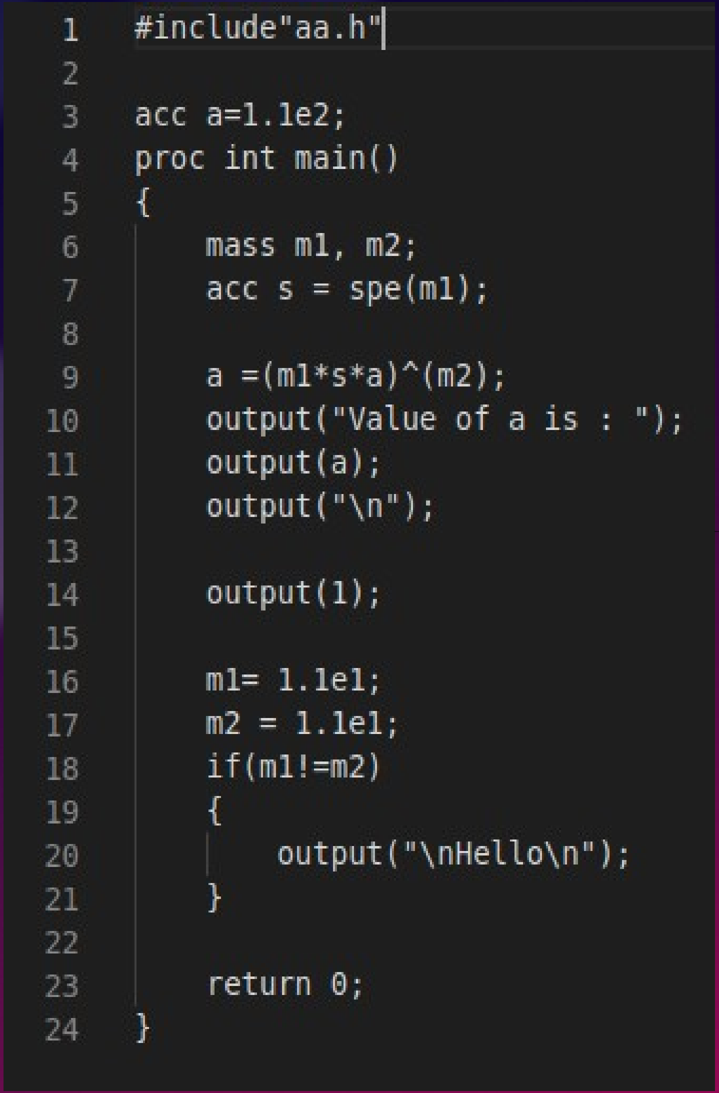
- For Transpiler, it involves converting/translating the semantically correct input code into another high-level programming language with appropriate headers defined for the new datatypes and new operators involved in the code.
- In our case, we convert:

.cos → .cpp

How to run Code generation phase?

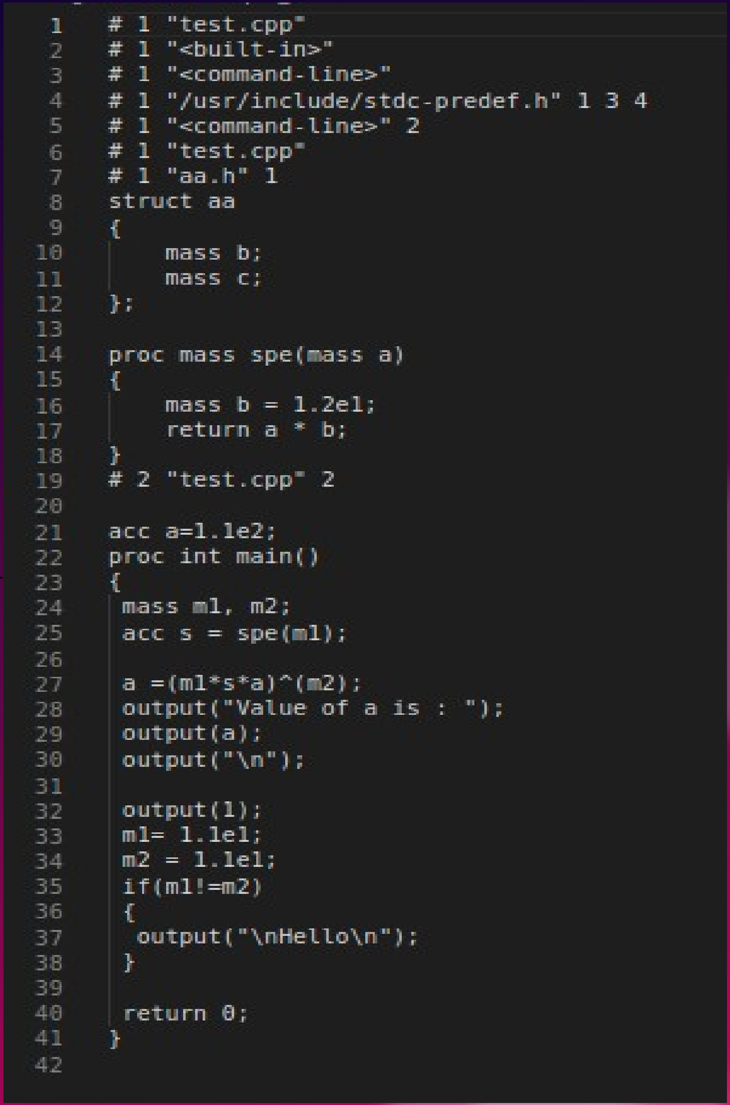
- **To run the code generation phase using Makefile on test cases, use command:** make
- **To run the code generation phase manually,**
  - flex lexer.l
  - bison -d -t parser.y
  - g++ -o testing_parser.out lex.yy.c parser.tab.c -lfl
  - cp testFile.cos testFile.cpp
  - g++ testFile.cpp -o PREPROCESSED_testFile.cpp.cos -E
  - rm testFile.cpp
  - ./testing_parser.out PREPROCESSED_testFile.cpp.cos

Image of working Code generation



```
#include"aa.h"

acc a=1.1e2;
proc int main()
{
    mass m1, m2;
    acc s = spe(m1);

    a =(m1*s*a)^(m2);
    output("Value of a is : ");
    output(a);
    output("\n");

    output(1);

    m1= 1.1e1;
    m2 = 1.1e1;
    if(m1!=m2)
    {
        output("\nHello\n");
    }

    return 0;
}
```

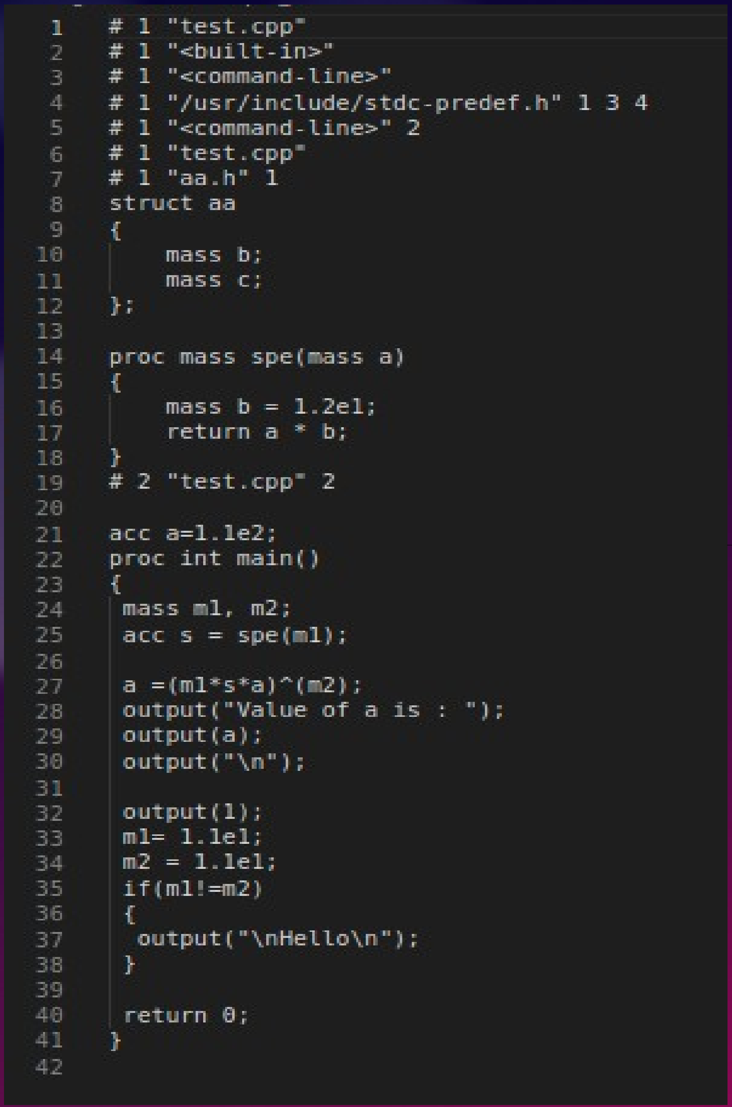
Source Code



```
# 1 "test.cpp"
# 1 "<built-in>"
# 1 "<command-line>"
# 1 "/usr/include/stdc-predef.h" 1 3 4
# 1 "<command-line>" 2
# 1 "test.cpp"
# 1 "aa.h" 1
struct aa
{
    mass b;
    mass c;
};

proc mass spe(mass a)
{
    mass b = 1.2e1;
    return a * b;
}
# 2 "test.cpp" 2

acc a=1.1e2;
proc int main()
{
 mass m1, m2;
 acc s = spe(m1);

 a =(m1*s*a)^(m2);
 output("Value of a is : ");
 output(a);
 output("\n");

 output(1);
 m1= 1.1e1;
 m2 = 1.1e1;
 if(m1!=m2)
 {
  output("\nHello\n");
 }

 return 0;
}
```

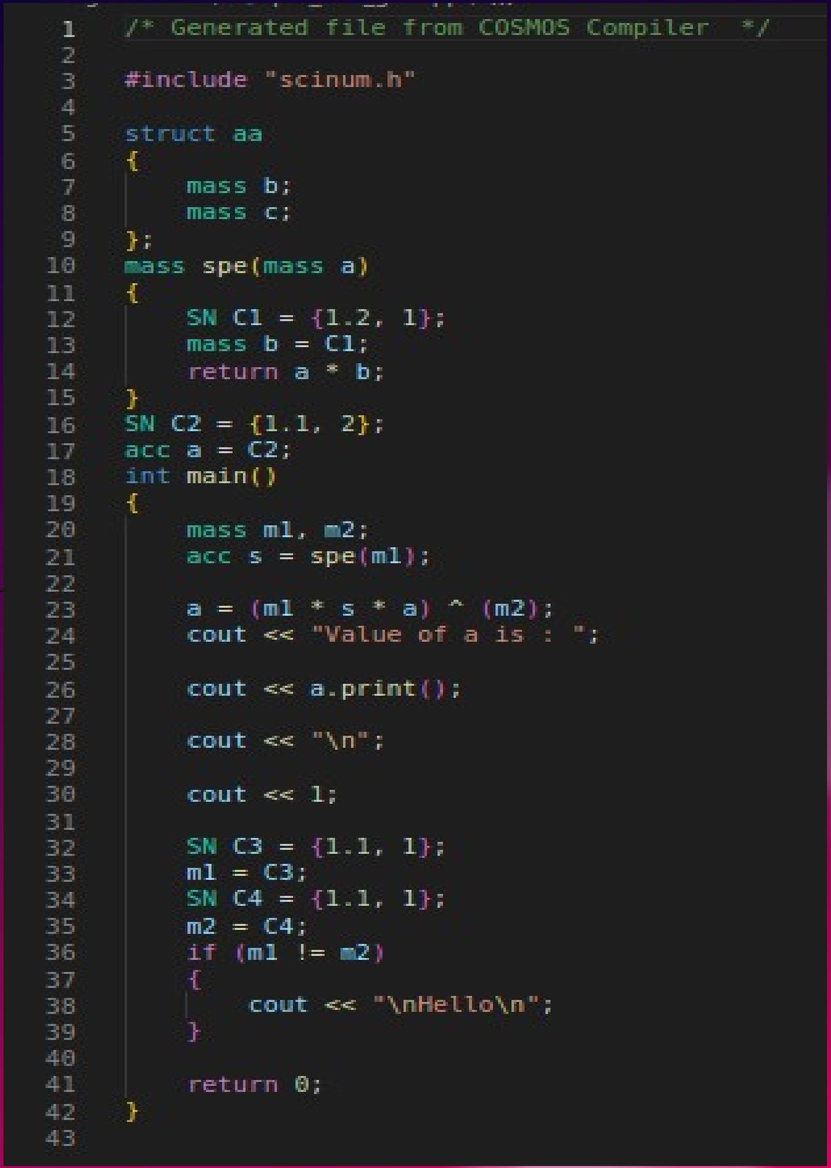
Preprocessed Code
(After Semantic, Before Codegen)

Image of working Code generation



```
# 1 "test.cpp"
# 1 "<built-in>"
# 1 "<command-line>"
# 1 "/usr/include/stdc-predef.h" 1 3 4
# 1 "<command-line>" 2
# 1 "test.cpp"
# 1 "aa.h" 1
struct aa
{
    mass b;
    mass c;
};

proc mass spe(mass a)
{
    mass b = 1.2e1;
    return a * b;
}
# 2 "test.cpp" 2

acc a=1.1e2;
proc int main()
{
 mass m1, m2;
 acc s = spe(m1);

 a =(m1*s*a)^(m2);
 output("Value of a is : ");
 output(a);
 output("\n");

 output(1);
 m1= 1.1e1;
 m2 = 1.1e1;
 if(m1!=m2)
 {
  output("\nHello\n");
 }

 return 0;
}
```

Preprocessed Code
(After Semantic, Before Codegen)



```
/* Generated file from COSMOS Compiler  */

#include "scinum.h"

struct aa
{
    mass b;
    mass c;
};
mass spe(mass a)
{
    SN C1 = {1.2, 1};
    mass b = C1;
    return a * b;
}
SN C2 = {1.1, 2};
acc a = C2;
int main()
{
    mass m1, m2;
    acc s = spe(m1);

    a = (m1 * s * a) ^ (m2);
    cout << "Value of a is : ";

    cout << a.print();

    cout << "\n";

    cout << 1;

    SN C3 = {1.1, 1};
    m1 = C3;
    SN C4 = {1.1, 1};
    m2 = C4;
    if (m1 != m2)
    {
        cout << "\nHello\n";
    }

    return 0;
}
```

After Codegen(Transpilation)