

# 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

```
1  #include"aa.h"|
2
3  acc a=1.1e2;
4  proc int main()
5  {
6      mass m1, m2;
7      acc s = spe(m1);
8
9      a =(m1*s*a)^(m2);
10     output("Value of a is : ");
11     output(a);
12     output("\n");
13
14     output(1);
15
16     m1= 1.1e1;
17     m2 = 1.1e1;
18     if(m1!=m2)
19     {
20         output("\nHello\n");
21     }
22
23     return 0;
24 }
```

Source Code

```
1  # 1 "test.cpp"
2  # 1 "<built-in>"
3  # 1 "<command-line>"
4  # 1 "/usr/include/stdc-predef.h" 1 3 4
5  # 1 "<command-line>" 2
6  # 1 "test.cpp"
7  # 1 "aa.h" 1
8  struct aa
9  {
10     mass b;
11     mass c;
12 };
13
14 proc mass spe(mass a)
15 {
16     mass b = 1.2e1;
17     return a * b;
18 }
19 # 2 "test.cpp" 2
20
21 acc a=1.1e2;
22 proc int main()
23 {
24     mass m1, m2;
25     acc s = spe(m1);
26
27     a =(m1*s*a)^(m2);
28     output("Value of a is : ");
29     output(a);
30     output("\n");
31
32     output(1);
33     m1= 1.1e1;
34     m2 = 1.1e1;
35     if(m1!=m2)
36     {
37         output("\nHello\n");
38     }
39
40     return 0;
41 }
42
```

Preprocessed Code  
(After Semantic, Before Codegen)

# Image of working Code generation

```
1 # 1 "test.cpp"
2 # 1 "<built-in>"
3 # 1 "<command-line>"
4 # 1 "/usr/include/stdc-predef.h" 1 3 4
5 # 1 "<command-line>" 2
6 # 1 "test.cpp"
7 # 1 "aa.h" 1
8 struct aa
9 {
10     mass b;
11     mass c;
12 };
13
14 proc mass spe(mass a)
15 {
16     mass b = 1.2e1;
17     return a * b;
18 }
19 # 2 "test.cpp" 2
20
21 acc a=1.1e2;
22 proc int main()
23 {
24     mass m1, m2;
25     acc s = spe(m1);
26
27     a =(m1*s*a)^(m2);
28     output("Value of a is : ");
29     output(a);
30     output("\n");
31
32     output(1);
33     m1= 1.1e1;
34     m2 = 1.1e1;
35     if(m1!=m2)
36     {
37         output("\nHello\n");
38     }
39
40     return 0;
41 }
42
```

Preprocessed Code  
(After Semantic, Before Codegen)

```
1  /* Generated file from COSMOS Compiler */
2
3  #include "scinum.h"
4
5  struct aa
6  {
7      mass b;
8      mass c;
9  };
10 mass spe(mass a)
11 {
12     SN C1 = {1.2, 1};
13     mass b = C1;
14     return a * b;
15 }
16 SN C2 = {1.1, 2};
17 acc a = C2;
18 int main()
19 {
20     mass m1, m2;
21     acc s = spe(m1);
22
23     a = (m1 * s * a) ^ (m2);
24     cout << "Value of a is : ";
25
26     cout << a.print();
27
28     cout << "\n";
29
30     cout << 1;
31
32     SN C3 = {1.1, 1};
33     m1 = C3;
34     SN C4 = {1.1, 1};
35     m2 = C4;
36     if (m1 != m2)
37     {
38         cout << "\nHello\n";
39     }
40
41     return 0;
42 }
43
```

After Codegen(Transpilation)