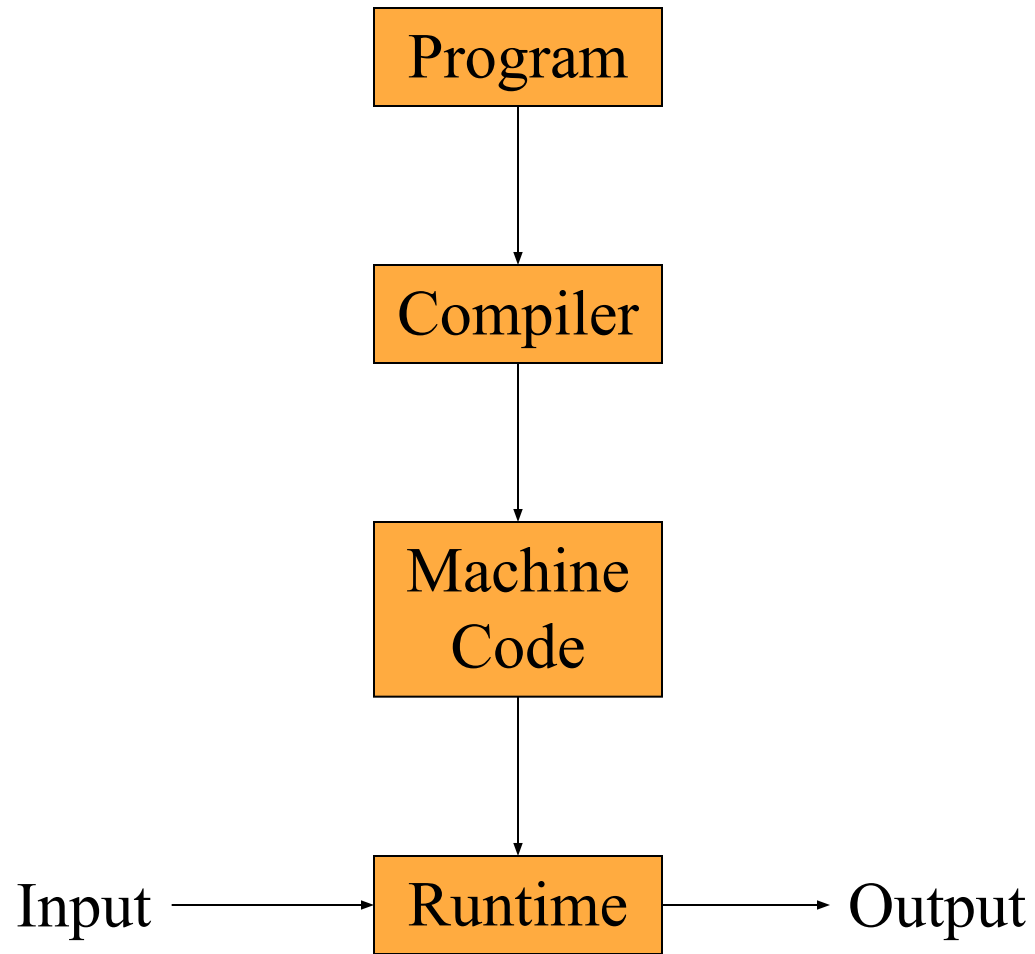


Introduction to Compilers

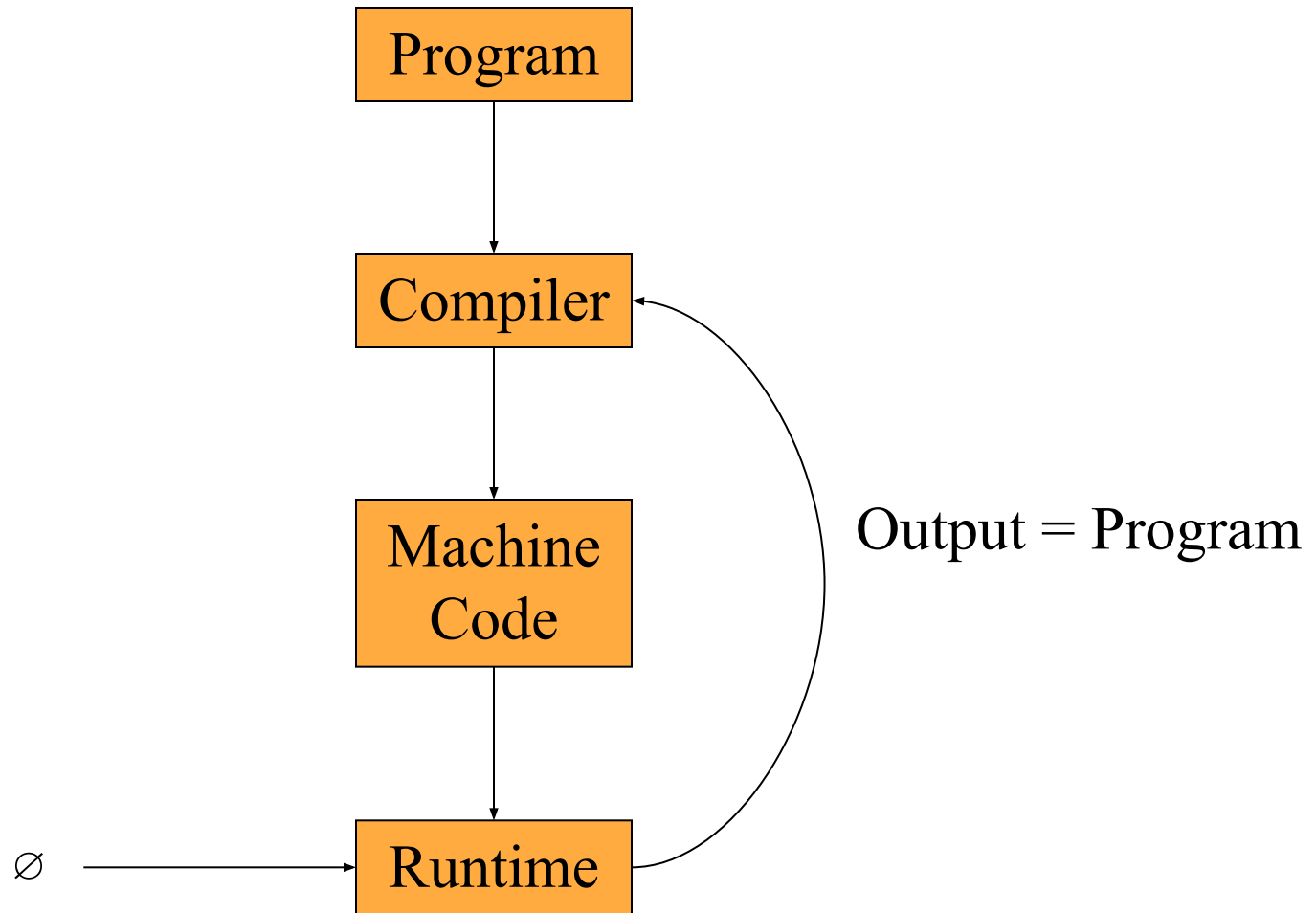
CMPT 379: Compilers

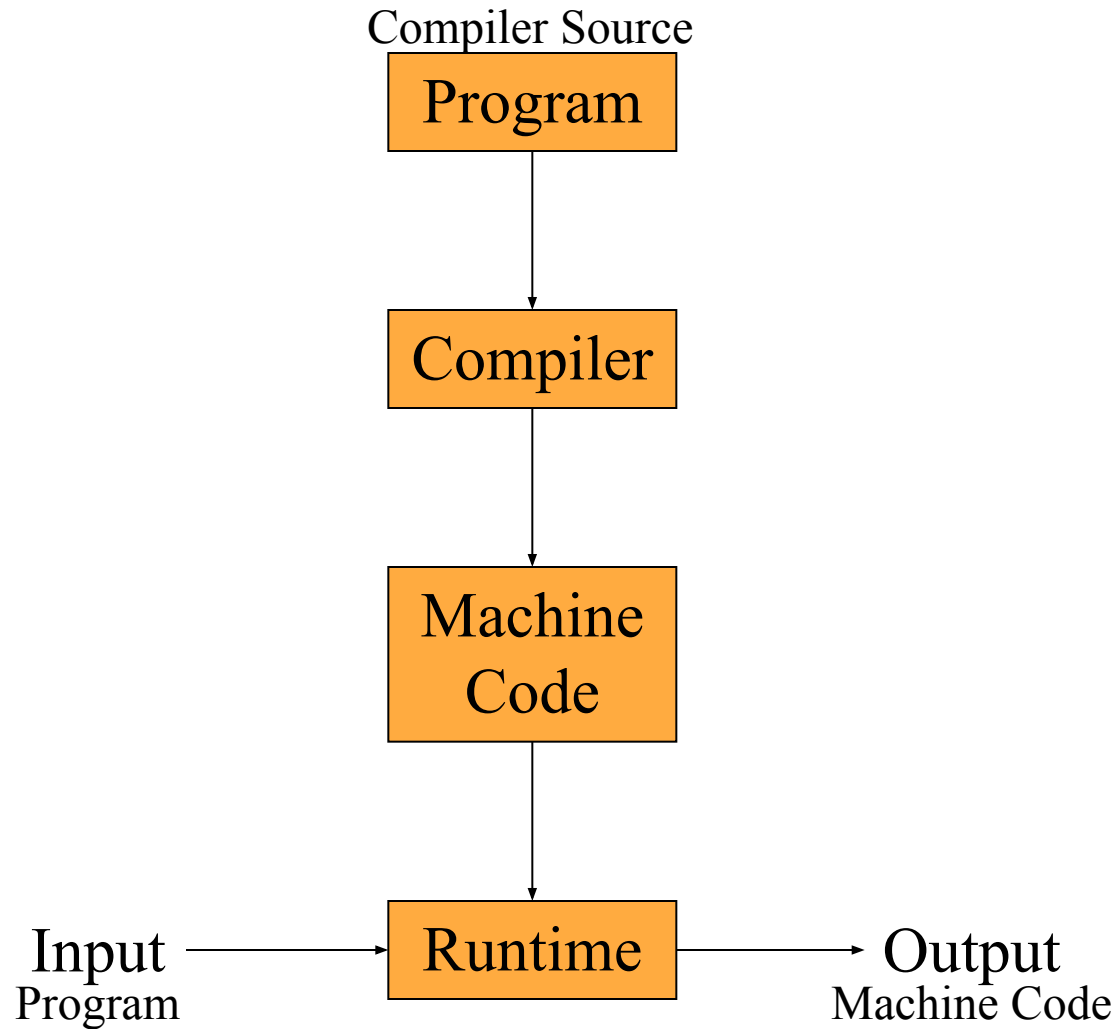
Instructor: Anoop Sarkar

anoopsarkar.github.io/compilers-class



```
main(){char *c="main(){char *c=%c%s%c;printf(c,34,c,34);}";  
printf(c,34,c,34);}
```





```
c = next();  
if (c == '\\') {  
    c = next();  
    if (c == 'n')  
        return('\\n');  
}
```

Compiler Source

Program

Compiler

ERROR: '\\n' not a valid character

Machine
Code

Input
Program

Runtime

Output
Machine Code

```
printf("hello world\\n")
```

```
c = next();  
if (c == '\\') {  
    c = next();  
    if (c == 'n')  
        return(10);  
}
```

Compiler Source

Program

Compiler

Machine
Code

Input
Program

Runtime

Output
Machine Code

```
printf("hello world\n")
```

```
c = next();  
if (c == '\\') {  
    c = next();  
    if (c == 'n')  
        return('\\n');  
}
```

Compiler Source

Program

New
Compiler

Machine
Code

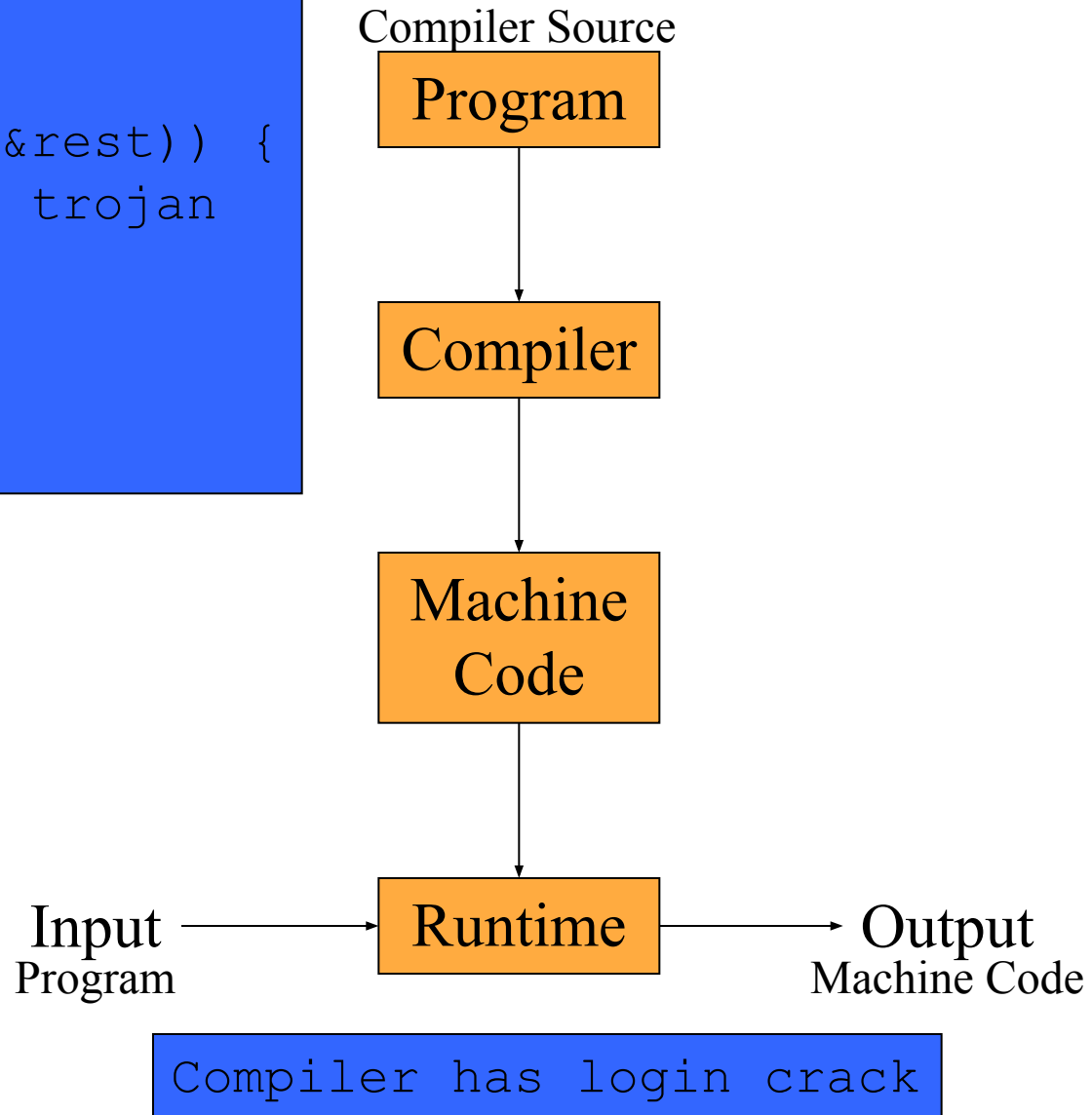
Input
Program

Runtime

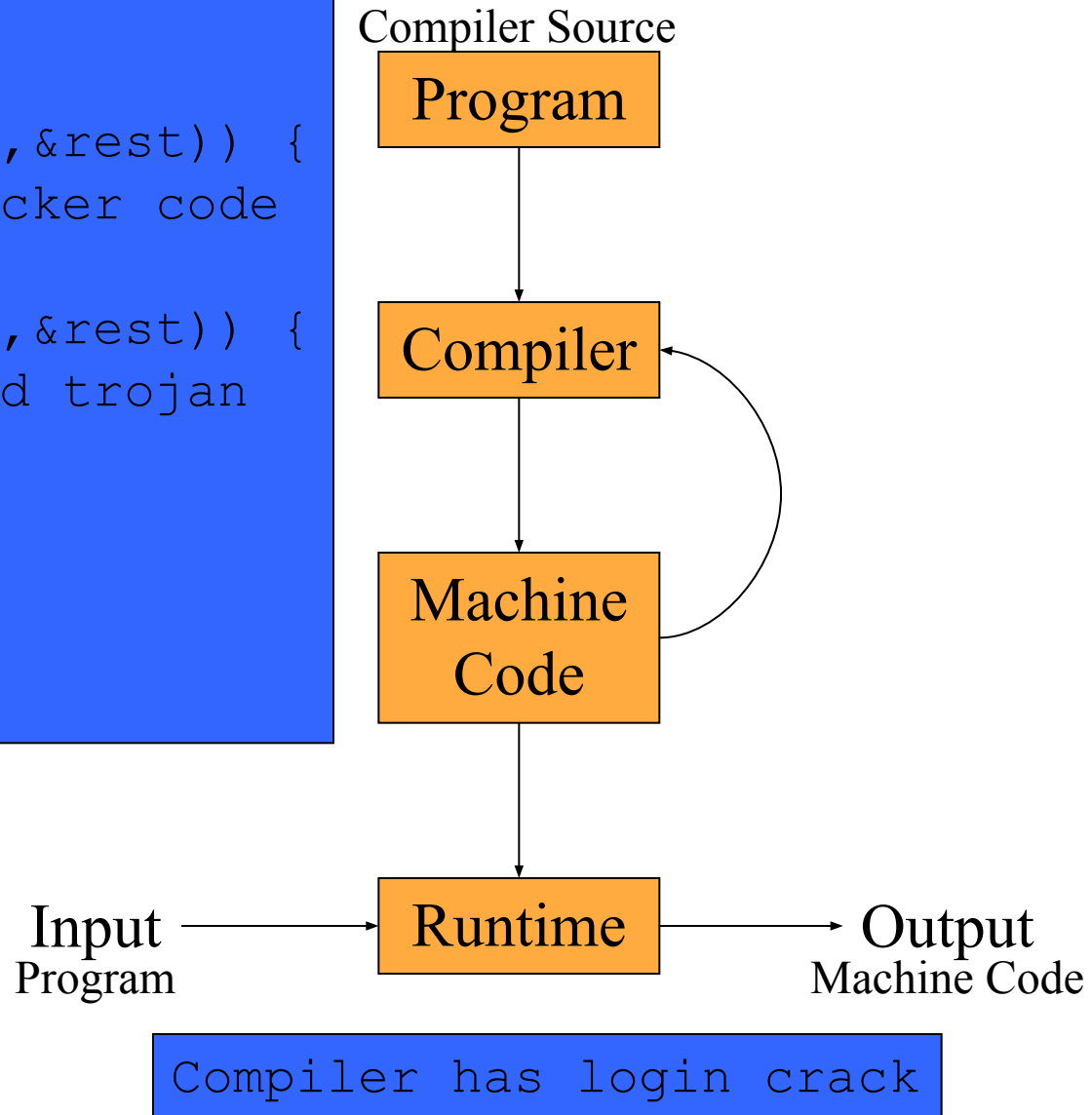
Output
Machine Code

```
printf("hello world\n")
```

```
compile(char *s)
{
    if(match(s,"login",&rest)) {
        // add root passwd trojan
        compile(rest);
    }
    ...
}
```




```
compile(char *s)
{
    if(match(s,"compile",&rest)) {
        // insert login cracker code
        compile("
        if(match(s,"login",&rest)) {
            // add root passwd trojan
            compile(rest);"");
        }
        compile(rest);
        ...
    }
}
```



```
compile(char *s)
{
    // standard compiler code
    // no login crack
    ...
}
```

Reflections on Trusting Trust,

Ken Thompson.

CACM 27(8), pp. 761-763, 1984.

