Proof Carrying Code (PCC):

- The diagram below illustrates the difference between Native Client and Proof Carrying Code.

**Good reasons why compiler generates the proof**

- Verification will be simpler to code when compiler generates the code
- Verification will be faster.
- There will be no or very little run time checking.
- Proof may be delivered from the source code.

Consider the following code

```c
int foo(int a)
{
    int b=a;
    if(b<0)
        b=-b;
    return b;
}
```

For the above code the correct post condition should be \( \text{foo}(x) \geq 0 \)

**Weakest Predicate:**

- \( \text{wp}(S,P) \) is the weakest predicate \( Q \), such that if \( Q \) is true before
executing S, then P will be true afterwards.

Eg:

\[
\begin{align*}
wp(x=x+1, \ x==7) \\
&\Rightarrow x==6 \ (\text{Pre condition}) \\
wp(x=e, \ P) \\
&= P[e/x] = x+1==7 \\
wp(x=x+1, \ y==5) \\
&\Rightarrow y==5 \ (\text{Pre condition}) \\
wp(x=x+1; \ x=2x, \ x==8) \\
&\Rightarrow 2(x+1) == 8 \\
&\Rightarrow x==3
\end{align*}
\]

Listed below are some of the common cases and their derivations

- \(wp(S1;S2, \ P)= wp(S1, wp(S2, \ P))\)
- \(wp(\text{if}(e) \ \text{then} \ S1; \ \text{else} \ S2, \ P)\)
- \(e \Rightarrow wp(S1,P)) \ ^\land \ (!e\Rightarrow wp(S2,P))\)
- \(wp(b=a; \ \text{if}(b<0) \ \text{then} \ b=-b \ \text{else} \ \{\}, \ b>=0)\)
  \[
  wp(b=a; \ (b<0 \Rightarrow -b>=0) \ ^\land \ (b>=0 \Rightarrow b>=0)) \\
  \Rightarrow (a<0 \Rightarrow -a >=0) \ ^\land \ (a>=0 \Rightarrow a>=0)
  \]

The steps followed in compiler & user are shown below

**COMPILER**

\[
\begin{align*}
\{\text{Pre Cond}\}, \ S, \ \{\text{Post Cond}\} \\
\text{Verification Condition(VC)} \\
\text{VC: Pre }\Rightarrow wp(S, \text{Post}) \\
\text{Theorem Prover} \\
\text{Proof}
\end{align*}
\]

**USER**

\[
\begin{align*}
\{\text{Pre Cond}\}, \ S, \ \{\text{Post Cond}\} \\
\text{VC Generator} \\
\text{Proof from compiler ,VC} \\
\text{Verifier} \\
\text{Valid / Invalid}
\end{align*}
\]