In[95]:=
```
{x, x + 1, x + 2, x^2} /. x → RandomInteger[100]
```
Out[95]=
```
{2, 3, 4, 4}
```

8/8

In[96]:=
```
{x, x + 1, x + 2, x^2} /. x :→ RandomInteger[100]
```
Out[96]=
```
{5, 59, 40, 441}
```

In[97]:=
```
f[x_] := x^2
```

In[98]:=
```
poly[n_Integer] := Graphics[Style[RegularPolygon[n], Orange]]
```

In[99]:=
```
Clear[f]
f[{x_, y_}] := {y, x}
```

In[101]:=
```
Clear[f]
f[{x_, y_}] := (x * y) / (x + y)
```

In[103]:=
```
Clear[f]
f[{x_, y_}] := {x + y, x - y, x / y}
```

In[105]:=
```
evenodd[0] = Red; evenodd[x_] := If[EvenQ[x] == True, Black, White]
```

In[106]:=
```
Clear[f]
f[{x_, y_, z_}] := If[x == 1, y + z, If[x == 2, y * z, If[x == 3, y^z]]]
```

On this one, he was meaning for you to use patterns rather than nested if statements, but of course both ways work.

In[108]:=
```
Clear[f]
f[0] = 1; f[1] = 1; f[n_Integer] := f[n - 1] + f[n - 2]
```

In[110]:=
```
animal[s_String] := Interpreter["Animal"][s]["Image"]
```

In[111]:=
```
nearwords[{s_String, n_Integer}] := Nearest[WordList[], s, n]
```