Extra Credit - Create your own problem to take advantage of using:
1. FindRoot[ ]
2. Solve [ ] and DSolve[ ]
3. /.

Create examples for both 1 and for 2 while using 3 in both.

The example below uses 1 and 3.

Two equations and two unknowns (x1 and x2)

Here I used := (SetDelayed) instead of the immediate = (Set). I do not know if it would make a difference in your use of the program.

In[1]:=
```
Clear[T, x2, x1, e1, e2]
e1 := 15178.993` - 10.926` T + 6000 x2^2 -
   8000 x1^2 + 8.314` T Log[1 - x2] - 8.314` T Log[1 - x1];
e2 := 7598.484` - 9.706` T + 6000 (1 - x2)^2 - 8000 (1 - x1)^2 +
   8.314` T Log[x2] - 8.314` T Log[x1];
solns := FindRoot[{e1 == 0, e2 == 0}, {x1, .1 }, {x2, .2 }];
tablesolns = Table[{x1, x2} = {x1, x2} /. solns, {T, 950, 1400, 25}]
(* Table of x2 and x2 solution *)
(* TableForm[tablesolns] *)
x1Table = Table[{x1, T} = {x1, T} /. solns, {T, 950, 1400, 25}]
(* Table of x1 solution and Temperatures *)
(* TableForm[x1Table] *)
x2Table = Table[{x2, T} = {x2, T} /. solns, {T, 950, 1400, 25}]
(* Table of x2 solution and Temperatures *)
(* TableForm[x2Table] *)
```

Out[5]=
```
{{0.531964, 0.788214}, {0.480746, 0.755355}, {0.435502, 0.720795}, {0.395205, 0.684296},
 {0.358854, 0.645721}, {0.325587, 0.605028}, {0.294704, 0.562269}, {0.265648, 0.517583},
 {0.237982, 0.471195}, {0.211361, 0.423393}, {0.185517, 0.374505}, {0.160241, 0.324877},
 {0.135373, 0.274841}, {0.11079, 0.224696}, {0.086398, 0.174697}, {0.0621287, 0.125047},
 {0.0379314, 0.0759011}, {0.0137701, 0.0273684}, {-0.0103804, -0.0204785}}
```

Out[6]=
```
{{0.531964, 950}, {0.480746, 975}, {0.435502, 1000}, {0.395205, 1025}, {0.358854, 1050},
 {0.325587, 1075}, {0.294704, 1100}, {0.265648, 1125}, {0.237982, 1150}, {0.211361, 1175},
 {0.185517, 1200}, {0.160241, 1225}, {0.135373, 1250}, {0.11079, 1275}, {0.086398, 1300},
 {0.0621287, 1325}, {0.0379314, 1350}, {0.0137701, 1375}, {-0.0103804, 1400}}
```

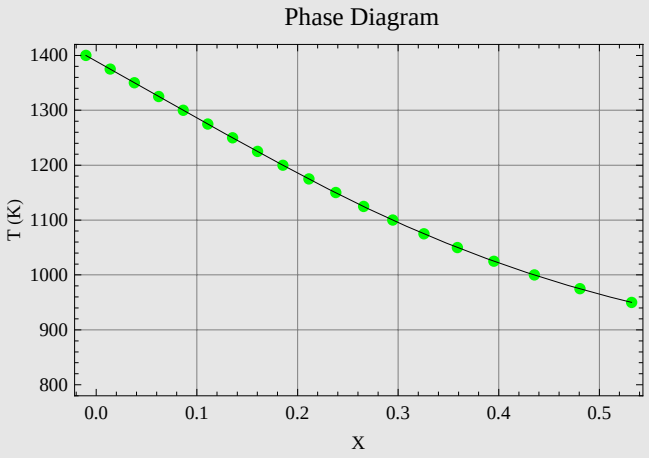
Out[7]=
```
{{0.788214, 950}, {0.755355, 975}, {0.720795, 1000}, {0.684296, 1025}, {0.645721, 1050},
 {0.605028, 1075}, {0.562269, 1100}, {0.517583, 1125}, {0.471195, 1150}, {0.423393, 1175},
 {0.374505, 1200}, {0.324877, 1225}, {0.274841, 1250}, {0.224696, 1275}, {0.174697, 1300},
 {0.125047, 1325}, {0.0759011, 1350}, {0.0273684, 1375}, {-0.0204785, 1400}}
```

Plotting the data in tables

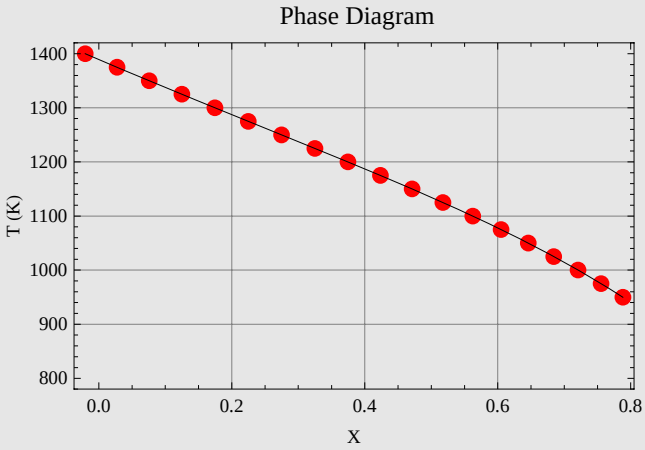
I added a line to ListPlot with the Epilog option. Below is another method.

In[8]:=
```
(*Plotting data*)
Clear[x1plot, x2plot]
x1plot = ListPlot[x1Table,
  AxesOrigin → {0, 800}, Frame → True, GridLines → Automatic,
  PlotStyle → {PointSize[0.02], RGBColor[0, 1, 0]},
  FrameLabel → {"X", "T (K)"}, PlotLabel -> " Phase Diagram",
  Epilog → {Line[x1Table], RGBColor[0, 1, 0]}]

x2plot = ListPlot[x2Table, AxesOrigin → {0, 800},
  Frame → True, GridLines → Automatic,
  PlotStyle → {PointSize[0.03], RGBColor[1, 0, 0]},
  FrameLabel → {"X", "T (K)"}, PlotLabel -> " Phase Diagram",
  Epilog → {Line[x2Table], RGBColor[1, 0, 0]}]

Show[{x1plot, x2plot}, PlotRange → {900, 1425},
 Epilog → {Line[{x1Table, x2Table}]}]
```

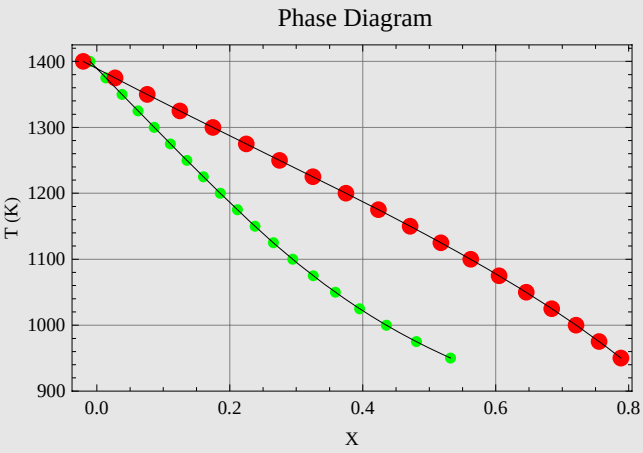
Out[9]=



Out[10]=



Out[11]=



Perhaps a little more efficient

Do the Table and calculations only once, then pull out columns with Part.
Instead of Epilog, I used a standard-package function DisplayTogether to make a ListPlot with points and line.

In[12]:=
```
Clear[T, x2, x1, e1, e2]
e1 := 15178.993` - 10.926` T + 6000 x2^2 -
   8000 x1^2 + 8.314` T Log[1 - x2] - 8.314` T Log[1 - x1];
e2 := 7598.484` - 9.706` T + 6000 (1 - x2)^2 - 8000 (1 - x1)^2 +
   8.314` T Log[x2] - 8.314` T Log[x1];
solns := FindRoot[{e1 == 0, e2 == 0}, {x1, .1 }, {x2, .2 }];

allsolns = Table[{x1, x2, T} = {x1, x2, T} /. solns, {T, 950, 1400, 25}]

x1Table = Part[allsolns, All, {1, 3}]
x2Table = Part[allsolns, All, {2, 3}]
```

Out[16]=
```
{{0.531964, 0.788214, 950}, {0.480746, 0.755355, 975},
 {0.435502, 0.720795, 1000}, {0.395205, 0.684296, 1025}, {0.358854, 0.645721, 1050},
 {0.325587, 0.605028, 1075}, {0.294704, 0.562269, 1100}, {0.265648, 0.517583, 1125},
 {0.237982, 0.471195, 1150}, {0.211361, 0.423393, 1175}, {0.185517, 0.374505, 1200},
 {0.160241, 0.324877, 1225}, {0.135373, 0.274841, 1250}, {0.11079, 0.224696, 1275},
 {0.086398, 0.174697, 1300}, {0.0621287, 0.125047, 1325}, {0.0379314, 0.0759011, 1350},
 {0.0137701, 0.0273684, 1375}, {-0.0103804, -0.0204785, 1400}}
```

Out[17]=
```
{{0.531964, 950}, {0.480746, 975}, {0.435502, 1000}, {0.395205, 1025}, {0.358854, 1050},
 {0.325587, 1075}, {0.294704, 1100}, {0.265648, 1125}, {0.237982, 1150}, {0.211361, 1175},
 {0.185517, 1200}, {0.160241, 1225}, {0.135373, 1250}, {0.11079, 1275}, {0.086398, 1300},
 {0.0621287, 1325}, {0.0379314, 1350}, {0.0137701, 1375}, {-0.0103804, 1400}}
```

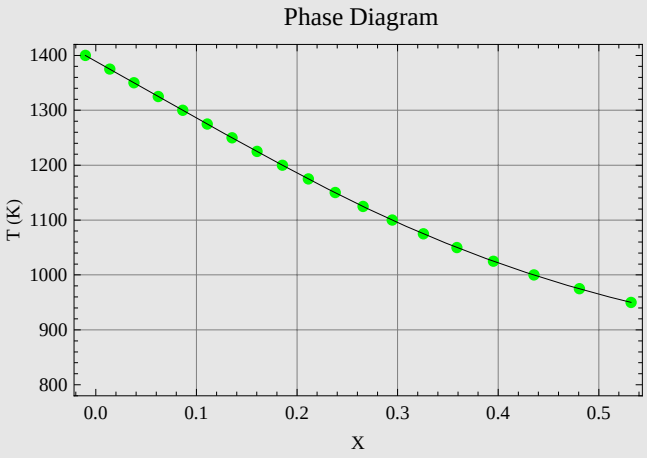
Out[18]=
```
{{0.788214, 950}, {0.755355, 975}, {0.720795, 1000}, {0.684296, 1025}, {0.645721, 1050},
 {0.605028, 1075}, {0.562269, 1100}, {0.517583, 1125}, {0.471195, 1150}, {0.423393, 1175},
 {0.374505, 1200}, {0.324877, 1225}, {0.274841, 1250}, {0.224696, 1275}, {0.174697, 1300},
 {0.125047, 1325}, {0.0759011, 1350}, {0.0273684, 1375}, {-0.0204785, 1400}}
```

Plotting the data in tables

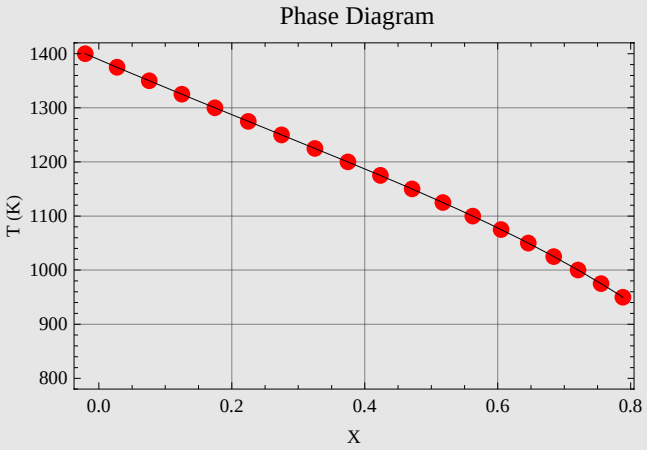
I added a line to ListPlot with the Epilog option. Below is another method.

In[19]:=
```
(*Plotting data*)
Clear[x1plot, x2plot]
x1plot = ListPlot[x1Table,
  AxesOrigin → {0, 800}, Frame → True, GridLines → Automatic,
  PlotStyle → {PointSize[0.02], RGBColor[0, 1, 0]},
  FrameLabel → {"X", "T (K)"}, PlotLabel -> " Phase Diagram",
  Epilog → {Line[x1Table], RGBColor[0, 1, 0]}]

x2plot = ListPlot[x2Table, AxesOrigin → {0, 800},
  Frame → True, GridLines → Automatic,
  PlotStyle → {PointSize[0.03], RGBColor[1, 0, 0]},
  FrameLabel → {"X", "T (K)"}, PlotLabel -> " Phase Diagram",
  Epilog → {Line[x2Table], RGBColor[1, 0, 0]}]

Show[{x1plot, x2plot}, PlotRange → {900, 1425},
 Epilog → {Line[{x1Table, x2Table}]}]
```

Out[20]=



Out[21]=



Out[22]=