Solution of $xy'' - y = 0$ by Mathematica

```
In[1]:= DSolve[{x * y''[x] - y[x] == 0}, y[x], x]
Out[1]= {{y[x] → -Sqrt[x] BesselI[1, 2 Sqrt[x]] C[1] + 2 Sqrt[x] BesselK[1, 2 Sqrt[x]] C[2]}}
```

Here we see the Mathematica code for solving the differential equation $xy'' - y = 0$ which we studied earlier using the Method of Frobenius, noting that the smaller of the two roots of the indicial equation failed to yield a solution. The top line shown is the input which must be typed exactly as shown. The output is normally written in mathematics as

$$y = \sqrt{x}\left(C_1 I_1(2\sqrt{x}) + C_2 K_1(2\sqrt{x})\right),$$

where we have modified C_1 to absorb the minus sign in the first summand and we have modified C_2 to absorb the factor 2.

Here is a good way to understand the output given by Mathematica. The fact that the modified Bessel functions I_1 and K_1 of order $\nu = 1$ are composed with $2\sqrt{x}$ means that we should replace the independent variable x with the new independent variable $t = 2\sqrt{x}$. And the outside factor $\sqrt{x}$ that multiplies what is inside the parentheses tells us that the dependent variable y should be replaced by a new dependent variable

$$u = \frac{y}{\sqrt{x}}.$$

The original equation should be written in terms of u, t, and the derivatives of u with respect to t, yielding a Modified Bessel equation, showing that in this example Mathematica's general solution is correct, albeit somewhat awkwardly expressed.

The work of changing variables is all a matter of careful use of the Chain Rule from calculus. For the problems shown in the homework, instead of using the complicated and rather limited formula given by Zill, you might as well use software such as Mathematica to find the correct changes of variables based on the form of the computer generated solution.

Your work should be to do the calculus with the Chain Rule to prove that Mathematica's solution is correct. In the near future, we will see an exercise in which Mathematica's solution is wrong and you will see how to recognize the error and how to correct it. In an engineering application, don't rely on software to solve a differential equation. You should always do the mathematics to confirm that the software generated solution is actually correct. Just consider, if you are responsible for solving such an equation as part of a $100 million dollar project and if you provide an incorrect solution, do you think the excuse that the software was in error will suffice? You need to always understand whether or not the solution is correct through your own reasoning.