```
import torch
import torch.nn as nn
import torch.nn.functional as F
```

```
# Create a Model Class that inherits nn.Module
class Model(nn.Module):
  # Input layer (4 features of the flower)
  #   --> Hidden Layer1 H1 (number of neurone)
  #   --> Hidden Layer H2 (n)
  #   --> output (which 3 classes of iris flower)

  def __init__(self, in_features = 4, h1 = 8, h2 = 9, out_features = 3):

    super().__init__() # Instantiate nn.Module (parent class)

    self.fc1 = nn.Linear(in_features, h1)
    self.fc2 = nn.Linear(in_features, h2)
    self.out = nn.Linear(h2, out_features)


  def forward(self, x):
    x = F.relu(self.fc1(x))
    x = F.relu(self.fc2(x))
    x = self.out(x)

    return x
```

```
# Pick a manual seed for randomization
torch.manual_seed(41)
# Create an instance of the model Model
model = Model()
```

```
```