In [2]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Activation,Dense
```

In [3]:

```
import pandas as pd
data=pd.read_csv(r"C:\Users\Anusha V\Downloads\heart1.csv")
data.head()
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

In [18]:

```
x = data.drop(columns=['age'])
y = data['cp']
```

In [19]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42
```

In [27]:

```
model = Sequential()
model.add(Dense(32, activation='relu', input_shape=(x_train.shape[1],)))
model.add(Dense(16, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

```

In [30]:

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(x_train, y_train, epochs=40, batch_size=32, validation_split=0.2)
```

```
Epoch 1/40
21/21 [==============================] - 3s 30ms/step - loss: -28.2103 - accuracy: 0.2149
- val_loss: -23.9889 - val_accuracy: 0.1402
Epoch 2/40
21/21 [==============================] - 0s 11ms/step - loss: -30.5557 - accuracy: 0.2058
- val_loss: -25.7636 - val_accuracy: 0.1402
Epoch 3/40
21/21 [==============================] - 0s 11ms/step - loss: -32.9630 - accuracy: 0.2210
- val_loss: -28.0729 - val_accuracy: 0.1463
Epoch 4/40
21/21 [==============================] - 0s 13ms/step - loss: -35.3603 - accuracy: 0.2241
- val_loss: -30.1004 - val_accuracy: 0.1402
Epoch 5/40
21/21 [==============================] - 0s 11ms/step - loss: -37.9226 - accuracy: 0.1829
- val_loss: -33.0718 - val_accuracy: 0.1341
Epoch 6/40
21/21 [==============================] - 0s 11ms/step - loss: -42.7039 - accuracy: 0.2088
- val_loss: -36.8453 - val_accuracy: 0.1402
Epoch 7/40
21/21 [==============================] - 0s 10ms/step - loss: -46.0811 - accuracy: 0.2302
```

In [31]:

```
test_loss, test_acc = model.evaluate(x_test, y_test)
print('Test accuracy:', test_acc)
```

```
7/7 [==============================] - 0s 7ms/step - loss: -298.5661 - accuracy: 0.2585
Test accuracy: 0.25853657722473145
```

In [ ]:

```

```

In [ ]:

```
```