In [4]:

```
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import img_to_array
from keras.applications.vgg16 import preprocess_input
from keras.applications.vgg16 import decode_predictions
from keras.applications.vgg16 import VGG16

image = load_img('C:/Users/Windows 10/Pictures/download.jpg', target_size=(224, 224
image = img_to_array(image)
image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
image = preprocess_input(image)
model = VGG16()
yhat = model.predict(image)
label = decode_predictions(yhat)
label = label[0][0]
print('%s (%.2f%%)' % (label[1], label[2]*100))
```

```
1/1 [==============================] - 1s 1s/step
golden_retriever (85.27%)
```

In [5]:

```
image = load_img('C:/Users/Windows 10/Pictures/download2.jpg', target_size=(224, 22

image = img_to_array(image)

image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))

image = preprocess_input(image)

model = VGG16()

yhat = model.predict(image)

label = decode_predictions(yhat)

label = label[0][0]

print('%s (%.2f%%)' % (label[1], label[2]*100))
```

```
WARNING:tensorflow:5 out of the last 5 calls to <function Model.make_predict_functio
n.<locals>.predict_function at 0x000001BFC0250CA0> triggered tf.function retracing.
Tracing is expensive and the excessive number of tracings could be due to (1) creati
ng @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3)
passing Python objects instead of tensors. For (1), please define your @tf.function
outside of the loop. For (2), @tf.function has reduce_retracing=True option that can
avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/gui
de/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/
function for  more details.
1/1 [==============================] - 1s 949ms/step
castle (31.92%)
```

In [6]:

```
image = load_img('C:/Users/Windows 10/Pictures/download1.jpg', target_size=(224, 22

image = img_to_array(image)

image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
image = preprocess_input(image)

model = VGG16()

yhat = model.predict(image)

label = decode_predictions(yhat)

label = label[0][0]

print('%s (%.2f%%)' % (label[1], label[2]*100))
```

```
WARNING:tensorflow:6 out of the last 6 calls to <function Model.make_predict_functio
n.<locals>.predict_function at 0x000001BFC0251A20> triggered tf.function retracing.
Tracing is expensive and the excessive number of tracings could be due to (1) creati
ng @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3)
passing Python objects instead of tensors. For (1), please define your @tf.function
outside of the loop. For (2), @tf.function has reduce_retracing=True option that can
avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/gui
de/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/
function for  more details.
1/1 [==============================] - 1s 1s/step
valley (45.85%)
```

```
In [ ]:
```