

Home Assignment:

- **Gender:** Represents the student's gender, categorized as 'Female' or 'Male'.
- **Age:** Indicates the age of the student in years.
- **Math Score:** The student's score in mathematics (0-100 scale), reflecting math proficiency.
- **Passing Status:** Indicates whether the student has passed ('Yes') or failed ('No'); this is the target variable for prediction.

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder

# Step 1: Create a DataFrame
data = {
    'Gender': ['Female', 'Male', 'Female', 'Male', 'Female',
              'Male', 'Female', 'Male', 'Female', 'Male'],
    'Age': [17, 16, 16, 17, 18, 18, 17, 16, 17, 18],
    'Math Score': [75, 50, 85, 60, 90, 40, 55, 70, 80, 65],
    'Passing Status': ['Yes', 'No', 'Yes', 'No', 'Yes',
                      'No', 'No', 'Yes', 'Yes', 'No']
}

df = pd.DataFrame(data)

# Display the table
print("Data Table:")
print(df)

# Step 2: Encode the categorical data
label_encoder = LabelEncoder()
df['Gender'] = label_encoder.fit_transform(df['Gender']) # Female = 0, Male = 1
df['Passing Status'] = label_encoder.fit_transform(df['Passing Status']) # No = 0, Yes = 1

# Step 3: Split the data into features and labels
X = df[['Gender', 'Age', 'Math Score']] # Features
y = df['Passing Status'] # Target variable (Passing Status)

# Step 4: Train the Decision Tree model
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X, y)

# Step 5: Predict new entry
def predict_passing_status(gender, age, math_score):
    # Create a DataFrame for the new entry to match feature names
    new_data = pd.DataFrame([[gender, age, math_score]], columns=['Gender', 'Age', 'Math Score'])
    prediction = clf.predict(new_data)
    return label_encoder.inverse_transform(prediction)[0]

# New entry for prediction
new_gender = 1 # Male (0 = Female, 1 = Male)
new_age = 17
new_math_score = 75

predicted_passing_status = predict_passing_status(new_gender, new_age, new_math_score)
print(f"\nThe predicted passing status for the new entry (Gender: {new_gender}, Age: {new_age}, Math Score: {new_math_score}) is: {predicted_passing_status}")
```

Data Table:

	Gender	Age	Math Score	Passing Status
0	Female	17	75	Yes
1	Male	16	50	No
2	Female	16	85	Yes
3	Male	17	60	No
4	Female	18	90	Yes
5	Male	18	40	No
6	Female	17	55	No
7	Male	16	70	Yes
8	Female	17	80	Yes
9	Male	18	65	No

The predicted passing status for the new entry (Gender: 1, Age: 17, Math Score: 75) is: Yes