In [1]:

```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
#Import the dataset
import sklearn.svm as svm
```

In [3]:

```
dataset = pd.read_csv(r'C:\Users\HP\Desktop\car_data.csv')
dataset.head()
```

Out[3]:

	User ID	Gender	Age	AnnualSalary	Purchased
0	385	Male	35	20000	0
1	681	Male	40	43500	0
2	353	Male	49	74000	0
3	895	Male	40	107500	1
4	661	Male	25	79000	0

In [4]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   User ID       1000 non-null   int64 
 1   Gender        1000 non-null   object
 2   Age           1000 non-null   int64 
 3   AnnualSalary  1000 non-null   int64 
 4   Purchased     1000 non-null   int64 
dtypes: int64(4), object(1)
memory usage: 39.2+ KB
```

In [5]:

```
dataset['Gender'].value_counts()
```

Out[5]:

```
Female    516
Male      484
Name: Gender, dtype: int64
```

In [6]:

```
#Converting gender values from object values to numerical values
#A sign Female to (0) and Male to (1)

convert = {"Gender": {"Female":0, "Male":1}}
```

In [7]:

```
dataset = dataset.replace(convert)
```

In [8]:

```
#dataset after convert the gender to numerical values
#data analysis
dataset
```

Out[8]:

	User ID	Gender	Age	AnnualSalary	Purchased
0	385	1	35	20000	0
1	681	1	40	43500	0
2	353	1	49	74000	0
3	895	1	40	107500	1
4	661	1	25	79000	0
...	...	...	...	...	...
995	863	1	38	59000	0
996	800	0	47	23500	0
997	407	0	28	138500	1
998	299	0	48	134000	1
999	687	0	44	73500	0

1000 rows × 5 columns

In [9]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
dataset['Purchased'] = le.fit_transform(dataset['Purchased'])
dataset.head(100)
```

Out[9]:

	User ID	Gender	Age	AnnualSalary	Purchased
0	385	1	35	20000	0
1	681	1	40	43500	0
2	353	1	49	74000	0
3	895	1	40	107500	1
4	661	1	25	79000	0
...	...	...	...	...	...
95	485	0	33	151500	1
96	960	1	45	75500	1
97	233	0	26	17000	0
98	191	1	30	87000	0
99	471	1	38	60500	0

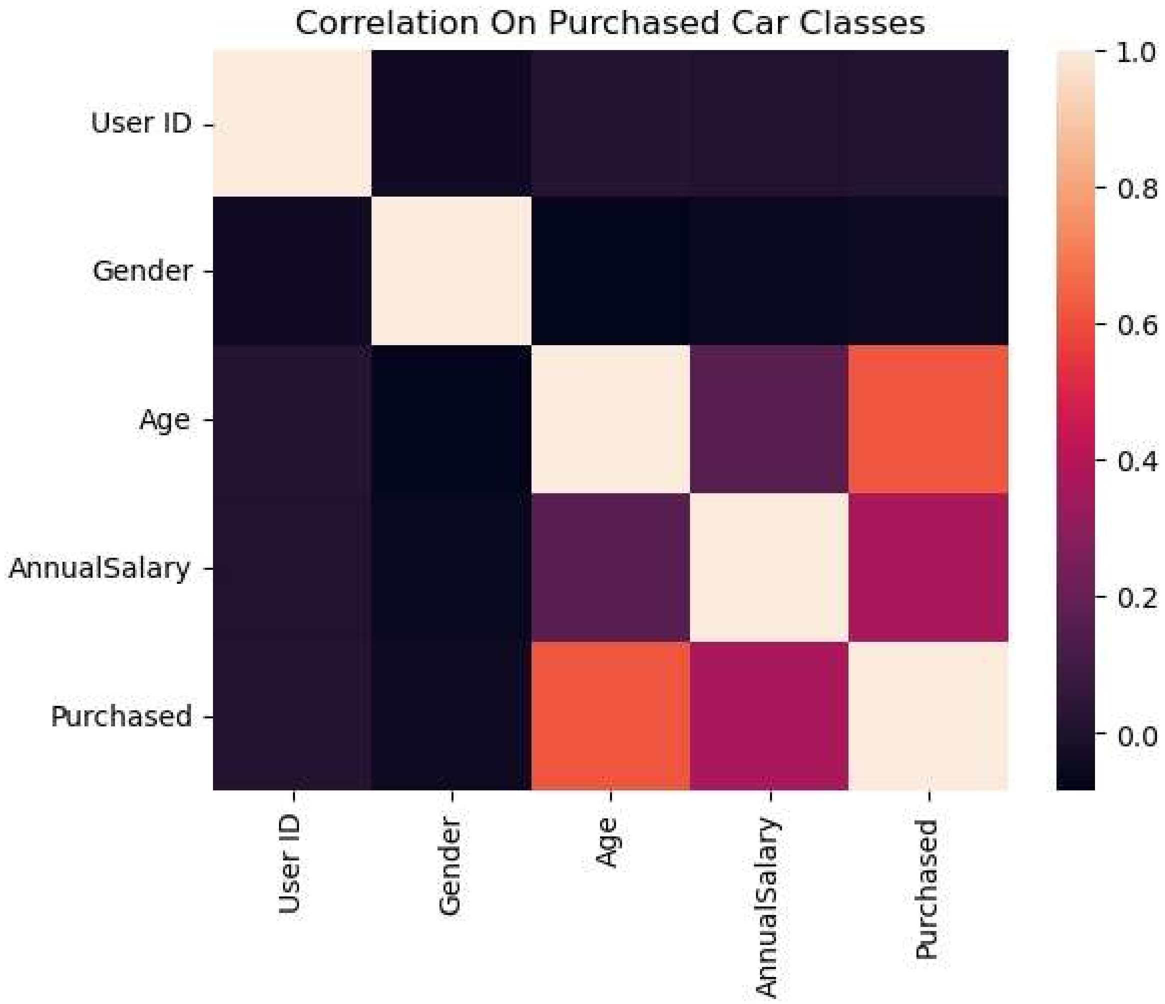
100 rows × 5 columns

In [10]:

```
plt.figure(1)
sns.heatmap(dataset.corr())
plt.title('Correlation On Purchased Car Classes')
```

Out[10]:

```
Text(0.5, 1.0, 'Correlation On Purchased Car Classes')
```



In [11]:

```
#Execute the following code to split the data into training and test sets
from sklearn.model_selection import train_test_split
```

In [13]:

```
#Data processing
X = dataset.drop(columns = ['Purchased'])
Y = dataset['Purchased']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25)
```

In [14]:

```
#Training and making predictions
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
classifier = DecisionTreeClassifier()
classifier.fit(X_train, Y_train)
y_pred = classifier.predict(X_test)
```

In [15]:

```
# Summary of the predictions made by the classifier
print(classification_report(Y_test, y_pred))
print(confusion_matrix(Y_test, y_pred))
# Accuracy score
from sklearn.metrics import accuracy_score
print('Accuracy is',accuracy_score(y_pred,Y_test))
```

```
              precision    recall  f1-score   support

           0       0.90      0.91      0.91       146
           1       0.87      0.87      0.87       104

    accuracy                           0.89       250
   macro avg       0.89      0.89      0.89       250
weighted avg       0.89      0.89      0.89       250

[[133  13]
 [ 14  90]]
Accuracy is 0.892
```

In [ ]: