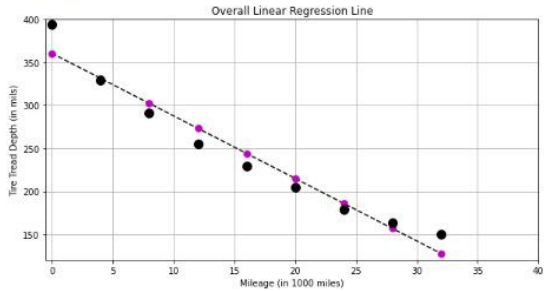
In [2]:
```
#Import all the necessary libraries
import numpy as np
import scipy.stats as st
import sklearn.linear_model as lm
import matplotlib.pyplot as plt
%matplotlib inline
```

In [14]:
```
#Create the necessary datasets
x = np.array([0, 4, 8, 12, 16, 20, 24, 28, 32])
y = np.array([394.33, 329.50, 291.00, 255.17, 229.33, 204.83, 179.00, 163.83, 150.33])

#Defining equation of the linear regression line
calculated_line_equation = 360.64 - (7.281 * x)

# We create the Linear Regression Model
linear_regression_model = lm.LinearRegression()
# We train the model on our training dataset.
linear_regression_model.fit(x[:, np.newaxis], y)
# Now, we predict points with our trained model.
sklearn_line_result = linear_regression_model.predict(x[:, np.newaxis])

print(f'Sklearn result: {sklearn_line_result}')
print(f'Own Calculation result: {calculated_line_equation}')
```

```
Sklearn result: [360.63666667 331.51416667 302.39166667 273.26916667 244.14666667
 215.02416667 185.90166667 156.77916667 127.65666667]
Own Calculation result: [360.64  331.516 302.392 273.268 244.144 215.02  185.896 156.772 127.648]
```

In [61]:
```
#Now I will plot the result of the trained linear model, and we obtain a regression line in blue below
fig, ax = plt.subplots(1, 1, figsize=(10, 5))
ax.plot(x, calculated_line_equation, '--k')
ax.plot(x, sklearn_line_result, '.m', ms=15)
ax.plot(x, y, 'ok', ms=10)
ax.set_xlim(-0.5, 40)
ax.set_ylim(120, 400)
ax.text(1, 4, r'Linear Regression Line Equation: $Y = 360.65 - 7.281X$', fontsize=10)
ax.set_title("Overall Linear Regression Line")
plt.xlabel("Mileage (in 1000 miles)")
plt.ylabel("Tire Tread Depth (in mils)")
plt.grid()
```



Linear Regression Line Equation: $Y = 360.65 - 7.281X$