```
function [theta, J_history] = gradientDescentMulti(X, y, theta, alpha,
 num_iters)
%GRADIENTDESCENTMULTI Performs gradient descent to learn theta
%   theta = GRADIENTDESCENTMULTI(x, y, theta, alpha, num_iters)
 updates theta by
%   taking num_iters gradient steps with learning rate alpha

% Initialize some useful values
m = length(y); % number of training examples
J_history = zeros(num_iters, 1);

for iter = 1:num_iters
    % ====================== YOUR CODE HERE ======================
    % Instructions: Perform a single gradient step on the parameter
 vector
    %               theta.
    %
    % Hint: While debugging, it can be useful to print out the values
    %       of the cost function (computeCostMulti) and gradient here.
    %
    predict = X*theta;
    for i = 1:size(X,2)
        theta(i) = theta(i) - alpha*1/m*sum((predict-y).*X(:,i));
    end
    % ============================================================

    % Save the cost J in every iteration
    J_history(iter) = computeCostMulti(X, y, theta);

end

end
```

```
Not enough input arguments.

Error in gradientDescentMulti (line 7)
m = length(y); % number of training examples
```

Published with MATLAB® R2019a