1. (a) Task: Model determines student major based on classes taken.
   (b) Input Features: classes, club activity, grades in class
   (c) Supervised with alumni, and semi-supervised for alumni that dropped out, and those not graduated yet.
   (d) Classification because the label (major) is more categorical than numerical.

2. The loss function will tell us how well the model is currently fitting to the one or more data points given. Empirical risk is the average loss under the empirical distribution of the training data and risk is the expected value of the empirical risk such that as the sample size approaches infinity, it would approach this expected value for risk. These concepts are related as they are all trying to estimate the same thing, being how well the model fits to the data.

3. A probabilistic model might be more helpful when trying to account for something where we do not understand the relationship well, i.e., an unsupervised model. For example, to continue on the height and weight model from the generalization video, if we instead wanted to predict sex based off height and weight, we would want to differentiate between male or female at given heights and weights where there would be probability distributions per sex at given height and weight inputs.

4. To check if our model is underfitting or overfitting, we can examine the loss function over time and check how it compares to the test set and dev set. If empirical risk is low, we are unlikely to have an under-fit. If the phi-regularization term is low, we are unlikely to have an over-fit model because the models are simple and in general are less capable of overfitting to the data.

5. Repeatedly seeing the test set is a problem because we will introduce bias. Your model can begin to overfit and it won't be unseen data to test against our generalization. Having bad generalization means the model won't be able to adapt to the new data.

6. Repeatedly seeing the dev set is not a bad thing so long as you keep the test set reserved to validate our model's predictions. Being able to see the dev set repeatedly will help us be able to see how our model is generalizing against more of our training sample. If we are not training with it, it should not alter the accuracy to predict how well the model will generalize for the test set and other unseen data.

7. An advantage of cross-validation would be getting an average test set generalization on the training data over the amount of folds to validate against the reserved test set. An advantage of having a dedicated dev set would be to prevent information leakage into the model during the training step. Also, a dedicated dev set can raise the question of overfitting to our validation data.

8. The loss on the dev/validation set before tuning would be approximately the same. After tuning we would expect the loss to be lower on the dev set because it (the model) has been tuned using the dev set. Although, we would not expect it to be much lower.

9. We are less likely to overfit if we regularize during training. The reason is because we are trying to optimize the balance between variance and bias, thus improving generalization error.

10. (a) $\frac{\partial f}{\partial w}(x, w) = x_1^2 + x_1 + x_2^2 + x_2 + 1$
    (b) $\frac{\partial f}{\partial x}(x, w) = 2x_1 w_1 + w_2 + 2x_2 w_3 + w_4$

(c) No, f is not a linear function of x, but it is a linear function of w. With respect to x, we have x_1^2 which is a polynomial and not linear by definition. With respect to w, imagine x's as constants there is no exponent above the degree of 1. All w's are to the power of 1 as well.