# Deep Learning Assignment 3:
# Denoising Diffusion Probabilistic Models

Due Date: **January 28th, 2026, 23:59**

**In this assignment:**

- You will implement Denoising Diffusion Probabilistic Models (DDPMs) from scratch.
- You will build a UNet architecture with time and text conditioning for image denoising.
- You will implement the forward noising process and reverse denoising process.
- You will train a text-conditioned diffusion model to generate emoji images.
- You will implement Classifier-Free Guidance for improved generation quality.

**Setup:** Same as previous assignments - use the free tier of Google Colab and upload the assignment folder to `My Drive/dl/assignments/assignment3/`. The notebook requires GPU acceleration for training.

## Denoising Diffusion Probabilistic Models (100 points)

**Notebook:** `DDPM.ipynb`

In this assignment, you will implement a complete Denoising Diffusion Probabilistic Model (DDPM) for text-conditioned emoji generation. DDPMs are a class of generative models that learn to reverse a gradual noising process, enabling the generation of high-quality images from pure noise.

You will:

- Implement the forward noising process (`q_sample`)
- Implement noise/image prediction functions to convert between predicted noise and clean images
- Build a UNet model with skip connections, time conditioning, and text conditioning
- Implement the denoising loss function (`p_losses`)
- Implement the reverse sampling process (`p_sample`)
- Implement Classifier-Free Guidance (`cfg_forward`)
- Generate emoji images conditioned on text prompts

The model will be trained on a dataset of emoji images with text descriptions, using CLIP embeddings for text conditioning.

## Submission Instructions

When you have completed the assignment, follow these steps to generate and submit your work:

### Step 1: Generate Submission Files

1. Ensure all code cells in the notebook have been executed and their outputs are visible
2. **IMPORTANT:** Manually save all `*.ipynb` and `*.py` files before proceeding
3. Open and run the `collect_submission.ipynb`. This will generate:
   - `a3_code_submission.zip`
   - `a3_inline_submission.pdf`
4. Download both files from your Google Drive to your local computer

### Step 2: Create Student Information File

Manually create a text file named `students.txt` containing both students' information in the following format:

```
Student1_Full_Name Student1_ID
Student2_Full_Name Student2_ID
```

For example:

```
John_Doe 123456789
Jane_Smith 987654321
```

**Note:** Use underscores (_) instead of spaces in the names, and ensure the student IDs match your official registration records.

### Step 3: Submit to Lemida

Submit the following three files to Lemida:

1. `a3_inline_submission.pdf`
2. `a3_code_submission.zip`
3. `students.txt`

**Good luck!**