**Assignment Documentation**

**OOP Concepts Applied:**

1. Classes and Objects:

- The application defines three classes: `Task`, `PriorityTask`, and `TaskList`.

- Each class represents a blueprint for creating objects. For example, `Task` objects represent individual tasks, while `TaskList` objects manage collections of tasks.

2. Encapsulation:

- Attributes of the classes are encapsulated using the `\_\_init\_\_` method to initialize them. For example, the `Task` class has attributes `title`, `description`, and `status`, which are set when a new task is created.

- The use of `self` allows each instance of the class to maintain its own state.

3. Inheritance:

- The `PriorityTask` class inherits from the `Task` class. This means that `PriorityTask` has all the attributes and methods of `Task`, but it also adds a new attribute, `priority`.

- The `super()` function is used in the `PriorityTask` constructor to call the parent class's constructor, ensuring that the `title` and `description` are initialized properly.

4. Method Overriding:

- The `\_\_str\_\_` method is overridden in both `Task` and `PriorityTask` classes to provide a custom string representation of the objects. This allows for a user-friendly display of task details when printed.

- For example, the `\_\_str\_\_` method in `PriorityTask` includes the priority level in its output.

5. Method Overloading:

- The `add\_task` and `add\_priority\_task` methods in the `TaskList` class demonstrate method overloading by allowing different ways of adding tasks. They can accept either just a title or both a title and a description.

- This flexibility allows users to create tasks with varying levels of detail.

6. Polymorphism:

- The `list\_tasks` method in the `TaskList` class utilizes polymorphism by calling the `\_\_str\_\_` method on each task object. Regardless of whether the task is a regular `Task` or a `PriorityTask`, the correct `\_\_str\_\_` method is invoked based on the object's type.

- This allows for a unified way to display tasks without needing to know their specific types.