```
import json

class Task:
    def __init__(self, title, description, category):
        self.title = title
        self.description = description
        self.category = category
        self.completed = False

    def mark_completed(self):
        self.completed = True

    def to_dict(self):
        return {
            'title': self.title,
            'description': self.description,
            'category': self.category,
            'completed': self.completed
        }

    @staticmethod
    def from_dict(data):
        task = Task(data['title'], data['description'],
data['category'])
```

```
        task.completed = data['completed']
        return task

    def __str__(self):
        status = "Completed" if self.completed else
"Pending"
        return f"Title: {self.title}\nDescription:
{self.description}\nCategory: {self.category}
\nStatus: {status}\n"

def save_tasks(tasks, filename='tasks.json'):
    with open(filename, 'w') as f:
        json.dump([task.to_dict() for task in tasks],
f, indent=4)

def load_tasks(filename='tasks.json'):
    try:
        with open(filename, 'r') as f:
            return [Task.from_dict(data) for data in
json.load(f)]
    except FileNotFoundError:
        return []

def view_tasks(tasks):
```

```
    if not tasks:
        print("\nNo tasks available.\n")
    else:
        for i, task in enumerate(tasks, 1):
            print(f"\nTask {i}:")
            print(task)

def add_task(tasks):
    title = input("Enter the task title: ")
    description = input("Enter the task description:
")
    category = input("Enter the task category (e.g.,
Work, Personal, Urgent): ")
    tasks.append(Task(title, description, category))
    print("\nTask added successfully!\n")

def mark_task_completed(tasks):
    if not tasks:
        print("\nNo tasks to mark as completed.\n")
        return
    view_tasks(tasks)
    task_num = int(input("Enter the task number
to mark as completed: ")) - 1
    if 0 <= task_num < len(tasks):
```

```
            tasks[task_num].mark_completed()
            print("\nTask marked as completed!\n")
        else:
            print("\nInvalid task number.\n")

def delete_task(tasks):
    if not tasks:
        print("\nNo tasks to delete.\n")
        return
    view_tasks(tasks)
    task_num = int(input("Enter the task number
to delete: ")) - 1
    if 0 <= task_num < len(tasks):
        tasks.pop(task_num)
        print("\nTask deleted successfully!\n")
    else:
        print("\nInvalid task number.\n")

def main():
    tasks = load_tasks()
```