```
from pydantic import BaseModel

from dataclasses import dataclass

from swarms import (

    create_yaml_schema_from_dict,

    YamlModel,

)



@dataclass

class TestDataClass:

    name: str

    age: int

    is_active: bool



class TestPydanticModel(BaseModel):

    name: str

    age: int

    is_active: bool



def test_create_yaml_schema_from_dict_dataclass():

    data = {"name": "Alice", "age": 30, "is_active": True}

    result = create_yaml_schema_from_dict(data, TestDataClass)

    expected_result = """

    name:
      type: str
      default: None
      description: No description provided
    age:
      type: int
      default: None
      description: No description provided
    is_active:
      type: bool
      default: None
      description: No description provided
    """
    assert result == expected_result


def test_create_yaml_schema_from_dict_pydantic():
    data = {"name": "Alice", "age": 30, "is_active": True}
    result = create_yaml_schema_from_dict(data, TestPydanticModel)
    expected_result = """
    name:
      type: str
      default: None
      description: No description provided
    age:
      type: int
      default: None
      description: No description provided
    is_active:
      type: bool
      default: None
      description: No description provided
    """
    assert result == expected_result



def test_create_yaml_schema_from_dict_regular_class():
    class TestRegularClass:
        def __init__(self, name, age, is_active):
            self.name = name
            self.age = age
            self.is_active = is_active

    data = {"name": "Alice", "age": 30, "is_active": True}
    result = create_yaml_schema_from_dict(data, TestRegularClass)
    expected_result = """
    name:
      type: str
      description: No description provided
    age:
      type: int
      description: No description provided
    is_active:
     type: bool
     description: No description provided
    """
    assert result == expected_result



class User(YamlModel):
    name: str
    age: int
    is_active: bool



def test_yaml_model():
    # Create an instance of the User model
    user = User(name="Alice", age=30, is_active=True)

    assert user.name == "Alice"
    assert user.age == 30
    assert user.is_active is True
```