

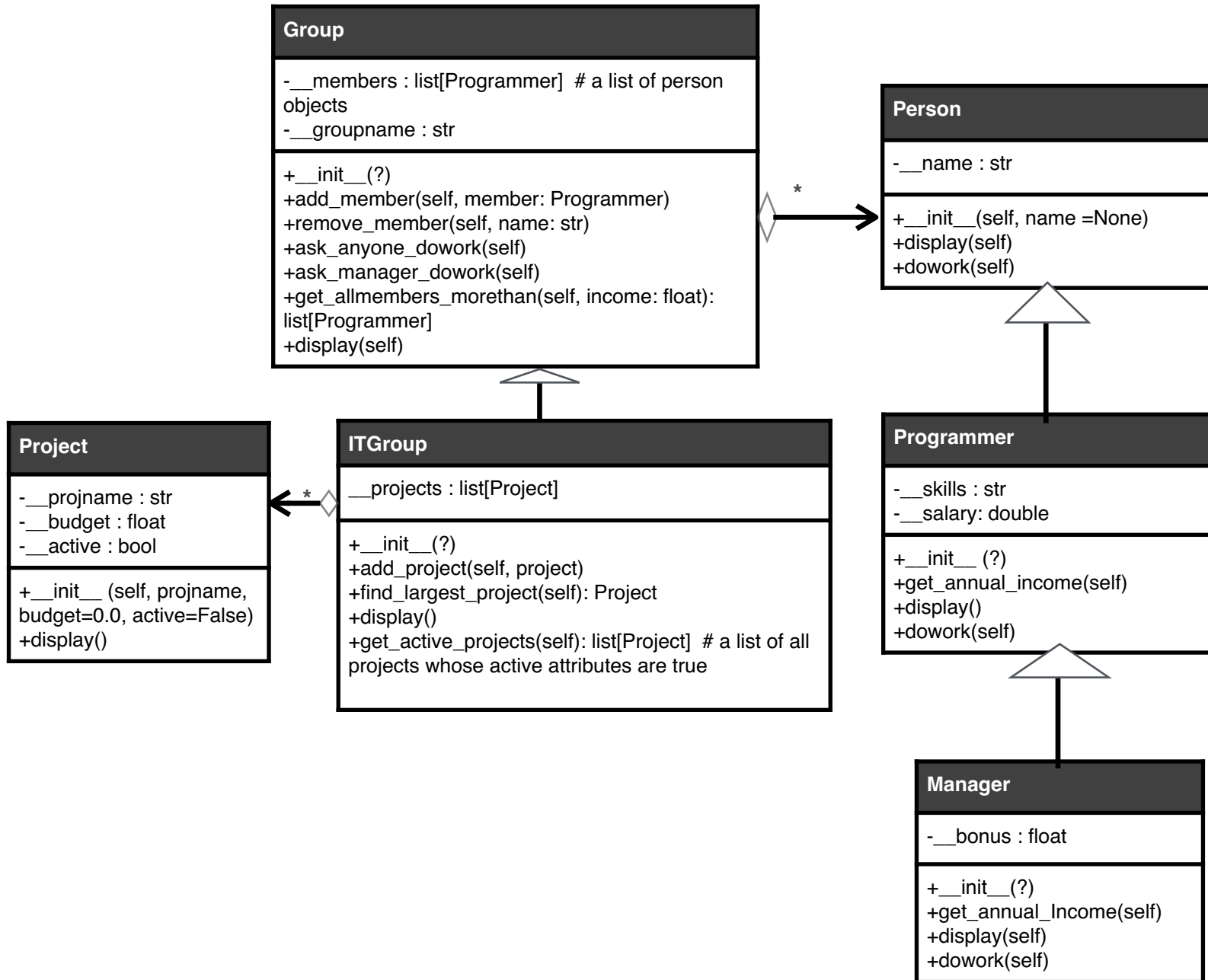
Question 1: Problem Description

Implement the following classes according to the class diagrams. And you should add appropriate getters and mutators using `@property` decorator if necessary.

You are required to define function parameters for the Group constructor (?), ITGroup constructor (?), Programmer constructor (?), and Manager constructor (?) in order to create objects of them properly.

Programmer constructor (?), and Manager constructor (?) in order to create objects of them properly.

- All classes need to implement display method.
- Annual Income = salary * 12 + bonus if any.
- After you implement all the classes and the methods specified in the class diagram, write a main method to create objects of the classes you defined, test their methods and print out their contents.



CS500 Lab - Inheritance

The below is a sample of main() method.

```
def main() -> None:
    p1: Programmer = Programmer("Lily", "C++, Java", 10000)
    p2: Programmer = Programmer("Judy", "Python, Java", 18000)
    m: Manager = Manager("Peter", "Management", 20000, 20000)
    proj1: Project = Project("MAX-5", 200000, True)
    proj2: Project = Project("FOX-4", 100000, False)
    proj3: Project = Project("FOX-XP", 500000, True)

    itgrp: ITGroup = ITGroup("ATX Group")
    itgrp.add_member(p1)
    itgrp.add_member(p2)
    itgrp.add_member(m)
    itgrp.add_project(proj1)
    itgrp.add_project(proj2)
    itgrp.add_project(proj3)
    itgrp.display()

    p3: Programmer = Programmer("Jone", "Python, Java", 1118000)
    itgrp.add_member(p3)

    itgrp.ask_anyone_dowork()
    print()
    itgrp.ask_manager_dowork()

    print("\nGet the largest project...")
    maxProj: Optional[Project] = itgrp.find_largest_project()
    if maxProj is not None:
        maxProj.display()

    print("\nGet the active projects...")
    projects: list[Project] = itgrp.get_active_projects()
    for proj in projects:
        proj.display()
        print()

    itgrp.display()
    itgrp.remove_member(p3.name)

    print("\nGet the members with high income...")
    members: list[Programmer] = itgrp.get_allMembers_morethan(200000)
    for member in members:
        member.display()
        print()

if __name__ == "__main__":
    main()
```

The expected output is:

The group has these members:
name = Lily

CS500 Lab - Inheritance

skills = C++, Java
salary = 10000

name = Judy
skills = Python, Java
salary = 18000

name = Peter
skills = Management
salary = 20000
bonus = 20000

The group has these projects:

projname = MAX-5
budget = 200000
active = True

projname = FOX-4
budget = 100000
active = False

projname = FOX-XP
budget = 500000
active = True

Programmer Lily is writing a program.

Programmer Judy is writing a program.

Manager Peter is supervising a team of programmers.

Programmer Jone is writing a program.

Manager Peter is supervising a team of programmers.

Get the largest project...

projname = FOX-XP
budget = 500000
active = True

Get the active projects...

projname = MAX-5
budget = 200000
active = True

projname = FOX-XP
budget = 500000
active = True

The group has these members:

name = Lily
skills = C++, Java
salary = 10000

name = Judy
skills = Python, Java
salary = 18000

name = Peter
skills = Management
salary = 20000
bonus = 20000

CS500 Lab - Inheritance

```
name = Jone  
skills = Python, Java  
salary = 1118000
```

The group has these projects:

```
projname = MAX-5  
budget = 200000  
active = True
```

```
projname = FOX-4  
budget = 100000  
active = False
```

```
projname = FOX-XP  
budget = 500000  
active = True
```

Get the members with high income...

```
name = Judy  
skills = Python, Java  
salary = 18000
```

```
name = Peter  
skills = Management  
salary = 20000  
bonus = 20000
```