

CS21 Practice QUIZ 6, Swarthmore College, Fall 2009

The first few questions refer in part to the following class:

```
class Person(object):

    def __init__(self, name):
        self.name = name
        self.friends = []

    def __str__(self):
        result = self.name + "'s Friends: "
        for f in self.friends:
            result = result + f.getName() + " "
        return result

    def hasFriend(self, name):
        for friend in self.friends:
            if friend.getName() == name:
                return True
        return False

    def getName(self):
        return self.name

    def getFriends(self):
        return self.friends

    def numFriends(self):
        return len(self.friends)

    def addFriend(self, friend):
        if not self.hasFriend(friend.getName()):
            #add a friend only if they aren't already a friend
            self.friends.append(friend)
        if not friend.hasFriend(self.name):
            friend.addFriend(self) #friendship is mutual
```

1. Given the following code fragment, show the value and type of the following expressions:

```
def mostFriends(ls):
    most = -1
    name = ""
    for person in ls:
        if person.numFriends() > most:
            most = person.numFriends()
            name = person.getName()
    return name
```

```
def main():
    p1 = Person("Harry")
    p2 = Person("Hermione")
    p3 = Person("Ron")

    p1.addFriend(p2)
    p1.addFriend(p3)

    print p1
    print p2
    b=p1.getFriends()
    print b[1]
    ls = [p1, p2, p3]
    print mostFriends(ls)
```

main()

Expression	Value	Type
-----	-----	-----
(1) print p1		
(2) p1.getName()		
(3) p1.getFriends()		
(4) p3.hasFriend("Harry")		
(5) mostFriends(ls)		
(6) b[1]		

2. Using the program from question (1), trace through the execution of the program showing both (1) the output and (2) the stack contents. Draw the stack right before the return statement in `mostFriends` is executed.
3. Write a class to represent a sports team object. Each object should have a team name and record the total number of wins and losses. Initially, each team should have 0 wins and 0 losses. Write code for methods `__init__`, `__str__`, `getName`, `getWins`, `getLosses`, `winGame`, `loseGame`, and `getWinningPercentage`. `winGame` and `loseGame` should increment the team's wins or losses, respectively, by one. `getWinningPercentage` should return the percentage of games won out of the total number of games played. If a team has played 0 games, report the winning percentage as `.000`.
4. Write a short `main` function to create one team object and test your methods. Show any output of your function.