1. I would not use seperate variables.

Instead I would define a class Student (Which will serve as the blueprint)

and then create objects

& store all objects in a list (Python) so I can manage the records as a whole.

eg:

```
class Student:
    def__init__(self, name, marks) :
    self.name = name
    self. marks = marks
    print(" adding new students in database")



S1= Student("Ani", 100)
print(s1. name, s1. marks)
.
.
.
.
```

& so on.

```
List = [ ]
List.append (s1, s2 ----)
```

2. Since it returns incorrect, i) I'd probably

check the return type and what it returns(avg or not),

ii)void might through error

has to be int/float.

ii) parameters inside the function

& their datatype

iii) main body of the program,

user input values etc.

3. Function is something we use to avoid
repetition in our code.
So for eg if we are calculating something
like a huge mathematical question
so it has several lines of code since the
calculation is pretty complex.
Maybe we are developing a software
we need to use the same
logic several times throughout the
Entire course/ lifetime of the project.
So will we write it everytime we need to
use that logic? Probably no since it's
tedious. So using a function we define
the logic once & call the function to
access the logic as and when needed.

4. Pseudocode in basic code structure
without so much focus on the syntaxes
It's like a bridge between algorithm & program.
If we write the pseudo code before actual code, it
helps us with efficient code writing
As it clears our algorithm also & logic step by step for a code.

1. Input 5 numbers
2. Store it in list (append)
3. initialize max to first element of list.
4. iterate through the entire length of the

List.

5. if element > max, replace max with

the element's value.

6. Loop runs as long as the largest element

isn't stored in max

7. once stored loop breaks and prints it

5. Team project

Version control with Github