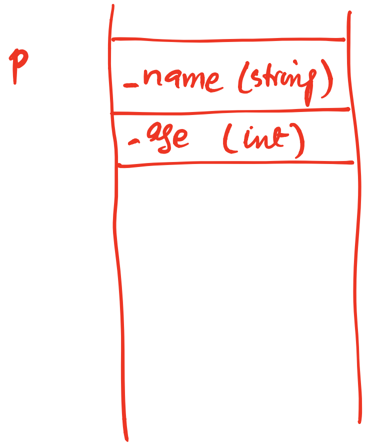
Inheritance

```
class Person {
   public:
   string _name;
   int _age;
};
```



p

_name (string)

_age (int)

Person p1;

Q. Does Person class has a default constructor? Yes

Q. Does it have an assignment operator? Yes

Shallow Copy

```
Person :: Person ()
{ }
```

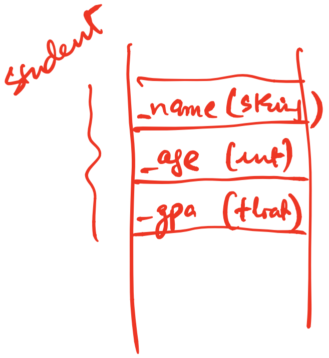
Goal: Student class : name, age, gpa

WITHOUT INHERITANCE

```
class Student
{
    public:
        string _name;
        int _age;
        float _gpa;

};

Student s1;
```

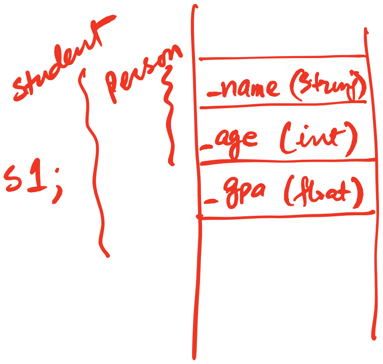


WITH INHERITANCE

```
class Student : public Person
{ public:
      float _gpa;

};

Student s1;
```



Person is referred to as "parent" class.

Student is referred to as "child" class.

"Child" class inherits members (class variables) and methods (class functions) of the "parent" class.

```
class Person
{
   public:
      Person (string name , int age)
      {
          _name = name;
          _age = age;
      }

      void print_info()
      {
         cout << _name << endl;
         cout << _age << endl;
      }

   protected:
      string _name;
      int _age;
};
```

```
class Student : public Person
{
   protected: float _gpa;
   Student(string name, int age, float gpa) : Person(name, age)
   {
      _gpa = gpa;
   }

}
```

```
Person  p1 ( "Jhon", 10);

p1. print_info();
```

```
Jhon
10
```

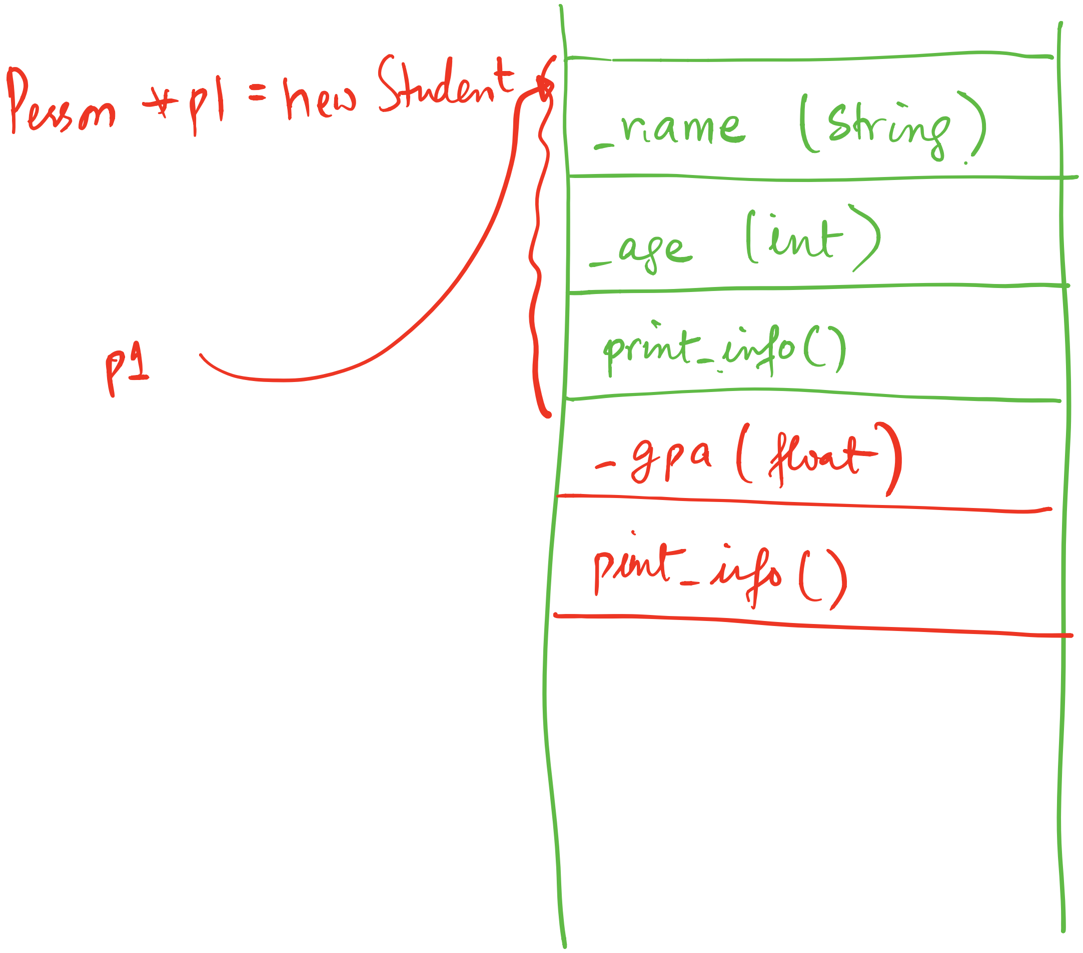
```
Student  s1 ("Jane" , "11", "4.3");
s1. print_info();
```

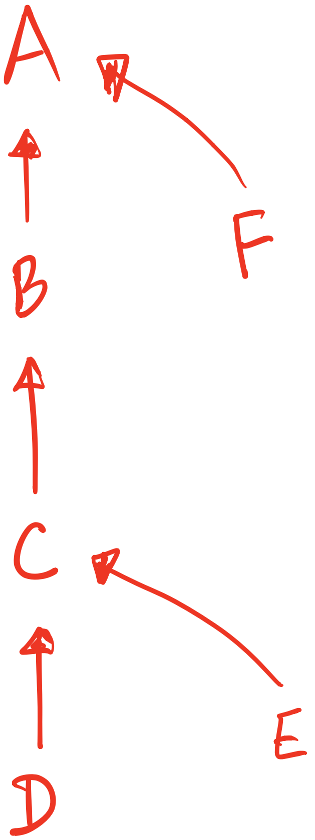
```
Jane
11
```

```
void  Student :: print_info ()
{
      cout << _name << endl;
      cout << _age << endl;
      cout << _gpa << endl;

}
```

```
Person :: print_info();
cout << _gpa << endl;
```


Person *p1 = new Student
p1
_name (string)
_age (int)
print_info()
_gpa (float)
pint_info()



```
A * a = new F;    ✓
B* b = new E;     ✓
B * b = new F;    ✗
D*d = new C;      ✗

F * f = new E;    ✗
```