**1-Usul**

**class Student:**

**def** printer(self):

print(self.**Pname**)

print(self.**Plname**)

O1**=Student()**

O1.**Pname**=’Olim’

O1.**Plname**=’Hakimov’

O1.printer()

**2-Usul**

**class Student:**

def**\_\_init\_\_(**self, name, lname**):**

self**.Pname=**name

self**.Plname=**lname

O1**=Student**(“Olim”, ”Hakimov”)

Print(O1**.Pname**)

Print(O2**.Plname**)

|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **O**(Object) | self**.Pname=**name | self**.Plname=**Iname |
| 1 | O1 | O1. **Pname=**Olim | 01.**Plname**=Hakimov |
| 2 | O2 | O2. **Pname=**Ilhom | 02.**Plname**=Asadov |

O1=O1 (Olim, Hakimov)

O2=O2 (Ilhom, Asadov)

O1. **Pname** = **Pname** O1 (Olim, Hakimov)=Olim

O1. **Plname** = **Plname** O1 (Olim, Hakimov)=Hakimov

O2. **Pname** = **Pname** O2 (Ilhom, Asadov)=Ilhom

O2. **Plname** = **Plname** O2 (Ilhom, Asadov)=Asadov

name, lname **–**O’zgaruvchilar **(Arguments)**

Pname, Plname **-** Xususiyatlar **(Properities)**

**3-Usul**

**class Student:**

def**\_\_init\_\_(**self, name, lname**):**

self**.Pname=**name

self**.Plname=**lname

def **Mname**(self):

print(self.**Pname**)

def **Mlname**(self):

print(self.**Plname**)

O1**=Student**(“Olim”, ”Hakimov”)

O1.**Mname()**

O1.**Mlname()**

|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **O**(Object**)** | self**.Pname=**name | self**.Plname=**Iname |
| 1 | O1 | O1.**Mname()=**Olim | 01.**Mlname()**=Hakimov |
| 2 | O2 | O2.**Mname()=**Ilhom | 02.**Mlname()**=Asadov |

O1=O1 (Olim, Hakimov)

O2=O2 (Ilhom, Asadov)

O1. **Mname()** = **Mname()** O1 (Olim, Hakimov)=Olim

O1. **Mlname()** = **Mlname()** O1 (Olim, Hakimov)=Hakimov

O2.**Mname()** = **Mname()** O2 (Ilhom, Asadov)=Ilhom

O2.**Mlname()** = **Mlname()** O2 (Ilhom, Asadov)=Asadov

name, lname **–** O’zgaruvchilar **(Arguments)**

Pname, Plname **-** Xususiyatlar **(Properities)**

Mname(), Mlname() **-** Metodlar **(Methods)**

|  |  |
| --- | --- |
| **N** | **C:\pyproject\oop\class1.py** |
| **1** | # bu yerda class hechqanday metodsiz berilgan.  **class** Comp:  x=”aple1”  o1=Comp()  print(o1.x)  # taqoslaymiz class ni print() funksiyasiga.  x=”aple1”  print(x) |
| **2** | # bu yerda class metod bilan amo, def \_\_init\_\_(self) funksiyasiz ishlatiladi.  **class** Comp  **def** config(self):  print(“aple2”)    o1=Comp()  o1.config() |
| **3** | # bu yerda class def \_\_init\_\_(self) funksiya bilan amo, medodsiz ishlatildi. Demak classlarda o’zgaruvchilarni odatiy x=”aple” ko’rinishda emas balkiy self.name=”aple” ko’rinishda saqlanar ekan. Taqoslab ko’rin print() funksiyasiga.    **class** Comp:  **def** \_\_init\_\_(self):  self.name=”aple3”    o1=Comp()  print(o1.name) |
| **4** | # bu yerda \_\_init\_\_(self) va metod bor va buni quyidagi funksiyaga taqoslaymiz, o’hshashlikni ko’rish uchun.  **class** Comp:  **def** \_\_init\_\_(self, name):  self.name=name  **def** config(self):  print(self.name)  o1=Comp(‘aple4’)  o1.config()  # bu fun() ni yuqoridagi class ga taqoslaymiz tuzilishini.  **def** fun(name):  print(name)  **fun**(‘aple4’) |
| **5** | # bu fun() funksiyani yuqoridagi class ga taqoslaymiz.  **def** fun(name):  print(name)  **fun**(‘aple5’) |
| **6** | # bitta metodi bor class  **class** Comp:  **def** config(self):  print(self.name)  o1=Comp()  o1.name=’aple6’  o1.config() |
| **7** | # o=[o1,o2, ….] ko’rinishda listga saqlab olamiz.  **class** Comp:  **def** \_\_init\_\_(self, name, age):  self.name=name  self.age=age  o1=[Comp(‘HP’, 20), Comp(‘aple’, 30)] |

|  |  |
| --- | --- |
| N | **C:\pyproject\oop\class2.py INNER and OUTER classes** |
| **1** | # bu yerda ichki va tashqi class larda bitadan metodlar bor, yaratilgan objectlarni tartibi muhim emas.  **class** Comp:  **def** show(self):  print(‘aple1’)  self.lap=self.Lap() # buni yozamasa ham bo’ladi.  **class** Lap:  **def** Show(self):  print(‘hp1’)  l1=Lap()  l1.show()  o1=Comp()  o1.show() |
| **2** | **class** Comp:  **def** \_\_init\_\_(self):  self.name=’aple2’  self.lap=self.Lap() # buni yozamasa ham bo’ladi.  **class** Lap:  **def** \_\_init\_\_(self):  self.lab=’hp2’  l1=Lap()  print(l1.lab)  o1=Comp()  print(o1.name) |
| **3** | **class** Comp:  **def**  show(self):  print(self.name)  self.lap=self.Lap() # buni yozamasa ham bo’ladi.    **class** Lap:  **def** show(self):  print(self.lab)  o1=Comp()  o1.name=’aple3’  l1=Lap()  l1.lab=’hp3’ |
| **4** | **class** Comp:  **def** **\_\_**init\_\_(self, name):  self.name=name  **def** show**(**self):  print(self.name)  **class** Lap:  **def \_\_**init\_\_(self,lab):  self.lab=lab  **def** show**(**self):  print(self.lab)    o1=Comp(‘aple4’)  o1.show()  l1=Lap(‘hp4’)  l1.show() |

|  |  |
| --- | --- |
| **N** | **C:\pyproject\oop\class3.py PASS** |
| 1 | **class** Person:  **def** show(self):  print(‘aple1’)  **class** Student(Person):  **pass**  o1=Student()  o1.show() |
| **2** | **class** Person:  **def** \_\_init\_\_(self):  self.name=’aple2’  **class** Student(Person):  **pass**  o1=Student()  print(o1.name) |
| **3** | **class** Person:  **def \_\_**init\_\_(self,name):  self.name=name  **def** show(self):  print(self.name)  **class** Student(Person):  **pass**  o1=Student(‘aple3’)  o1.show() |
| 4 | **class** Person:  **def** show(self):  print(self.name)  **class** Student(Person):  **pass**  o1=Student()  o1.name=’aple4’  o1.show() |
| **5** | **class** Person:  **def \_\_**init\_\_(self):  self.name=’aple5’  **def** show(self):  print(self.name)  **class** Student(Person):  **pass**  o1=Student()  o1.show() |
| **6** | # bu yerda odiyParent class berilgan bo’lib, biz quyida **PASS** dan foydalanib uning qiymatini child clasga o’tkazamiz.  **class** Person:  **def** \_\_init\_\_(self, name, lname):  self.name=name  self.lname=lname  **def** print\_name**(**self.name, self.lname)  o1=Person**(‘**elyor’,’dusnazarov’)  o1.print\_name()  # (Parent +child) classlar, bu **PASS** parent classni barcha hosalarini child clasga o’tkazib beradi.  **class** Person:  **def** \_\_init\_\_(self, name, lname):  self.name=name  self.lname=lname  **def** print\_name(self.name, self.lname)  **class** Student(Person):  **pass**  o1=Student(‘elyor’,’dusnazarov’)  o1.print\_name() |
| **7** | # Agar **parent** class **…\oop\parclass.py** da saqlangan bo’lsin  **class** Person:  **def** \_\_init\_\_(self, name, lname):  self.name=name  self.lname=lname  **def** print\_name(self.name, self.lname)  o1=Person(‘elyor’,’dusnazarov’)  o1.print\_name()  # Agar **child** class **…\oop\chilclass.py** da saqlangan bo’lsin  **from**  parclass import **Person**    **class** Student(Person):  **pass**    o1=Student(‘elyor’,’dusnazarov’)  o1.print\_name() |
| **8** | # Agar **parent** class **…\oop\parclass.py** da saqlangan bo’lib, object yaratilmagan bo’lsin.  **class** Person:  **def** \_\_init\_\_(self, name, lname):  self.name=name  self.lname=lname    **def** print\_name**(**self.name, self.lname)  # Agar **child** class **…\oop\chilclass.py** da saqlangan bo’lsin  **from** parclass import **Person**    **class** Student(Person):  **pass**    o1=Student(‘elyor’,’dusnazarov’)  o1.print\_name() |
|  |  |

|  |  |
| --- | --- |
| N | **C:\pyproject\oop\class4.py** |
| 1 | # agar ikkita o’zaro bog’lanmagan parent class berilgan bo’lsa ularni natijasi ham birin ketinlikda chiqadi.  **class** A:  **def** \_\_init\_\_(self):  print(‘A in init’)  **class** B:  **def** \_\_init\_\_(self):  print(‘B in init’)  a1= A()  b1=B() |
| 2 | # agar child class ni \_\_init\_\_() funksiyasini ishlatsek parent classni \_\_init\_\_() funksiyasi ishlamaydi.  **class** A:  def \_\_init\_\_(self):  print(‘A in init’)  **class** B(A):  **def** \_\_init\_\_(self**)**:  print(‘B in init’)  b1=B() |
| 3 | # agar child class ni \_\_init\_\_() funksiyasi bo’lmasa parent class ishlovradi.  **class** A:  **def** \_\_init\_\_(self):  print(‘A in init’)  class B(A):  **def** show(self):  print(‘B’)  b1=B()  b1.show() |
| 4 | # A class ni hosalarini A().\_\_init\_\_() funksiyadan foydalanib B child clasga o’tkazdik.  **class** A:  def \_\_init\_\_(self):  print(‘A in init’)  **class** B(A):  **A().\_\_**init**\_\_()**  **def** \_\_init\_\_(self):  print(‘B in init’)  b1=B() |
| 5 | **class** A:  def \_\_init\_\_(self,name):  self.name=name  **def** show(self):  print(self.name)  **class** B(A):  **def** \_\_init\_\_(self,lab):  **A.** \_\_init\_\_(self):    b1= B(‘aple’)  b1.show() |

|  |  |
| --- | --- |
| **N** | **C:\pyproject\oop\class5.py** |
| 1 | **class** Person:  **def** \_\_init\_\_(self,name,lname):  self.name=name  self.lname=lname  **def** show\_per(**self**):  print(self.name,self.lname)  **class** Student(Person):  **def** \_\_init\_\_(self,name,lname,year):  **super()**.\_\_init\_\_(name,lname)  self.year=year  x=Student(‘Mike’,’Jons’,2010)  x.show\_per()  print(x.year) |
| 2 | **class** Person:  def \_\_init\_\_(self,name,lname):  self.name=name  self.lname=lname  **def** show\_per(self):  print(self.name,self.lname)  **class** Student(Person):  **def** \_\_init\_\_(self,name,lname):  **super().**\_\_init\_\_(name,lname)  self.year=2014  x=Student(‘Mike’,’Jons’)  x.show\_per()  print(x.year) |
| 3 | **class** Person:  def \_\_init\_\_(self):  print(‘aple’)  **class** Student(Person):  **def** \_\_init\_\_(self,name,lname):  super().\_\_init\_\_()  self.name=name  self.lname=lname  **def** show\_stu(self):  print(self.name,self.lname)  x=Student(‘Mike’,’Jons’)  x.show\_stu() |
| 4 | **class** Person:  def show\_per(self):  print(‘aple1’)  **class** Student(Person):  **def** \_\_init\_\_(self,name,lname):  self.name=name  self.lname=lname  **def** show\_stu(self):  print(self.name,self.lname)  x=Student(‘Mike’,’Jons’)  x.show\_stu()  x.show\_per() |
| 5 | **class** Person:  **def** \_\_init\_\_(self):  self.year=2019  self.age=6  **class** Student(Person):  **def** \_\_init\_\_(self,name,lname):  super().\_\_init\_\_()  self.name=name  self.lname=lname  **def** show\_stu(self):  print(self.name,self.lname)  x=Student(‘Mike’,’Jons’)  x.show\_stu()  print(x.year,x.age) |
| 6 | **class** Person:  **def** \_\_init\_\_(self):  self.year=2019  self.age=6  **class** Student(Person):  **def** \_\_init\_\_(self):  super().\_\_init\_\_()  **def** show\_stu(self):  print(self.name,self.lname)  x=Student()  x.name=’Tom’  x.lname=’Jons’  x.show\_stu()  print(x.year,x.age) |
| 7 | **class** Person:  **def** \_\_init\_\_(self,name,lname):  self.name=name  self.lname=lname  **def** show\_per(self):  print(self.name,self.lname)  **class** Student(Person):  **def** \_\_init\_\_(self,name,lname,year,age):  **super().**\_\_init\_\_(name,lname)  self.year=year  self.age=age  **def** show\_stu(self):  print(self.year,self.age)  b=Student(‘Mike’,’Jons’,2019,3)  b.show\_per()  b.show\_stu() |