|  |
| --- |
| Employee\_details |
| +Name-string  +Pay\_rate-float  +Hours\_worked – float  +Salary – float |
|  |

Class diagram:

Pseudocode:

Define class “ employee\_details”:

Define attributes “name” ,”pay\_rate“, “hours\_worked “, ”salary “

Define layout when printed \_\_str\_\_:

“Name:” name

“pay rate:” pay\_rate

“hours worked:” hours\_worked

“salary:” salary

Get employee name

repeat until valid:

input name

if name is invalid ask again

else accept

get employee pay rate

repeat until valid:

input pay rate

if rate is less than 0 ask again

if data type is not valid ask again

else accept

repeat until valid:

input hours worked

if hours is less than 0 ask again

if data type is not valid ask again

else accept

repeat until valid:

input salary

if salary is less than 0 ask again

if data type is not valid ask again

else accept

Save the name, payrate , hours worked, salary as user

Output user

Actual code:

class employee\_details:

    def \_\_init\_\_(self,Name,Pay\_rate,Hours\_worked,Salary): #define all the atributes of the worker

        self.Name=Name

        self.Pay\_rate=Pay\_rate

        self.Hours\_worked = Hours\_worked

        self.Salary = Salary

    def \_\_str\_\_(self): #format how the data is presented when called for

        return(f"""Name: {self.Name}

Pay rate: {self.Pay\_rate}

Hours worked: {self.Hours\_worked}

Salary: {self.Salary}""")

while True:#get the name as long as its not an empty space

    name = input("What is the name of the employee? ").strip()

    if name.isnumeric():

        print("name must be a string")

    elif name:

        break

    else:

        print("Name cannot be empty.")

while True: # dont take the input if it is anything other than a float and makes sures its not a negative

    try:

        Pay\_rate=float(input("what is the employees pay rate?"))

        if Pay\_rate < 0:

            print("Pay rate cannot be negative!")

        else:

            break

    except ValueError:

        print("input has to be a number!")

while True: # dont take the input if it is anything other than a float and makes sures its not a negative

    try:

        Hours\_worked=float(input("how many hours do they work?"))

        if Hours\_worked < 0:

            print("Pay rate cannot be negative!")

        else:

            break

    except ValueError:

        print("input has to be a number!")

while True: # dont take the input if it is anything other than a float and makes sures its not a negative

    try:

        Salary=float(input("what is the employees salary?"))

        if Salary < 0:

            print("Pay rate cannot be negative!")

        else:

            break

    except ValueError:

        print("input has to be a number!")

user=employee\_details(name,Pay\_rate,Hours\_worked,Salary) #assign the attributes to the user using the class

print(user)#display the details in the requested format

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated