

# ASSIGNMENT – 7.2

NAME : P.HANSINI REDDY  
ROLL NO : 2403A510D5  
BATCH : 01  
COURSE : AI ASSISTED CODING

## TASK-1:

### PROMPT:

Python snippet that shows a syntax error.

**a = 10**

**if a = 10:**

**print("Equal")**

fix the syntax error in the conditional statement?

### CODE:

[1]  
✓ 0s

```
a = 10
if a == 10:
    print("Equal")
```

↔ Equal

## EXPLANATION:

In Python, you need to use `==` for comparison in conditional statements, not `=`.

## TASK 2:

### PROMPT:

There's an off-by-one error in the loop.

```
def sum_upto_n(n):  
    total = 0  
    for i in range(1, n):  
        total += i  
    return total, fix the error
```

### CODE:

```
def sum_upto_n(n):  
    total = 0  
    for i in range(1, n + 1):  
        total += i  
    return total  
  
# Example usage:  
print(sum_upto_n(5)) # This should print 15 (1+2+3+4+5)
```

 15

## TASK – 3:

### PROMPT:

i have attribute error in this code , give me correct code  
class User:

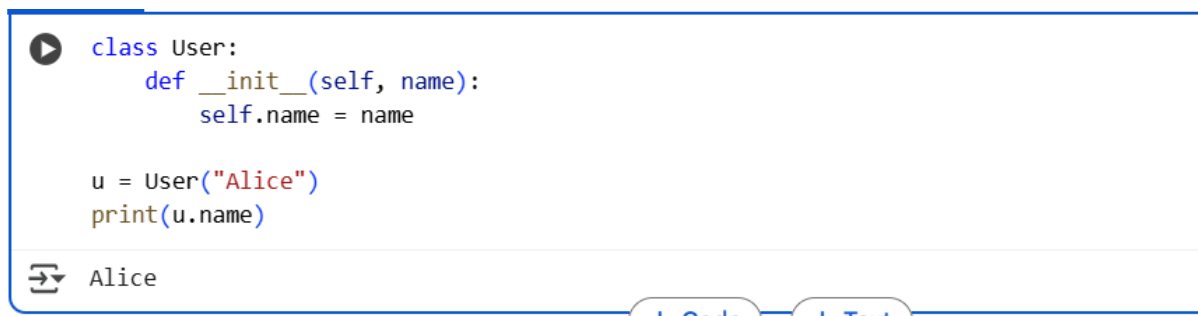
```
def __init__(self, name):
```

```
    self.name = name
```

```
u = User("Alice")
```

```
print(u.getName())
```

## CODE:



```
class User:
    def __init__(self, name):
        self.name = name

u = User("Alice")
print(u.name)
```

Alice

## EXPLANATION:

the AttributeError. The User class has an attribute called name, but there isn't a method named getName()

## TASK - 4:

### PROMPT:

```
class Car:
```

```
def start():
```

```
print("Car started")
```

```
mycar = Car()
```

mycar.start() , identify the error and give me correct code

## CODE:

```
class Car:
    def start(self):
        print("Car started")

mycar = Car()
mycar.start()
```

Car started

## EXPLANATION:

The error in the code is that the start method within the Car class is missing the self parameter. In Python, when you define a method within a class that will be called on an instance of that class, you need to include self as the first parameter.

## TASK – 5:

### PROMPT:

The grade levels don't have any logic give me correct code for this,

```
grade_student(score):
```

```
if score < 40:
```

return "A"


elif score < 70:

return "B"

else:

return "C"

## CODE:

```
 def grade_student(score):  
    if score >= 90:  
        return "A"  
    elif score >= 80:  
        return "B"  
    elif score >= 70:  
        return "C"  
    elif score >= 60:  
        return "D"  
    else:  
        return "F"  
  
# Example usage:  
print(grade_student(95)) # Expected: A  
print(grade_student(85)) # Expected: B  
print(grade_student(75)) # Expected: C  
print(grade_student(65)) # Expected: D  
print(grade_student(55)) # Expected: F
```



A  
B  
C  
D  
F