> created by Luis S. Flores for ITESM Contact: Luis_SFU@tec.mx
> August/2025

Conditional Control Structures

A conditional is a **control structure**.

Sequential structure: the computer runs instructions line by line, top to bottom, without decisions.

Control structure: the computer decides which lines to run, based on a True/False test (a condition).

Operators You Put Inside Conditions

Relational (comparisons)

Operator	Meaning	Example	Result Example
`==`	equal to	`x == y`	`5 == 5` → True
`!=`	not equal to	`x != y`	`4 != 5` → True
`>`	greater than	`x > y`	`7 > 5` → True
`<`	less than	`x < y`	`3 < 5` → True
`>=`	greater than or equal	`x >= y`	`5 >= 5` → True
`<=`	less than or equal	`x <= y`	`4 <= 5` → True

Note: `<>` is not valid in Python 3. Use `!=`.

Logical (combine or flip conditions)

Operator	Meaning	Example	Result Example
`and`	both must be True	`x>5 and y<10`	`7>5 and 3<10` → True
`or`	at least one is True	`x>5 or y<10`	`2>5 or 3<10` → True
`not`	flip True/False	`not(x>5)`	`not(7>5)` → False

Membership (in a collection)

Operator	Meaning	Example	Result Example
`in`	left value is inside right	`x in [1,2,3]`	`2 in [1,2,3]` → True
`not in`	left value is not inside	`x not in "abc"`	`'z' not in "abc"` → True

Identity (same object in memory)

Operator	Meaning	Example	Note
`is`	is the same object	`a is b`	Use `==` for value equality
`is not`	is not the same object	`a is not b`	Identity ≠ equality

Conditional Structures with Short Python Examples

In [53]:
```python
# Simple if
x = 10
if x > 5:
    print("x > 5")  # prints because 10 > 5
```

```
x > 5
```

In [54]:
```python
# if – else
x = 3
if x > 5:
    print("x > 5")
else:
    print("x <= 5")  # prints because 3 <= 5
```

```
x <= 5
```

In [55]:
```python
# if – elif – else
x = 7
if x > 10:
    print("x > 10")
elif x > 5:
    print("5 < x <= 10")  # prints
else:
    print("x <= 5")
```

```
5 < x <= 10
```

In [56]:
```python
# Nested if
x = 12
if x > 5:
    if x % 2 == 0:
        print("x > 5 and even")  # prints
    else:
        print("x > 5 and odd")
```

```
x > 5 and even
```

In [57]:
```python
# while with a condition
count = 3
print("Countdown")
while count > 0:
    print(count)
    count = count - 1
print("Lift off!")
```

```
Countdown
3
2
1
Lift off!
```

In [58]:
```python
# "Switch"-style in Python (match/case, Python 3.10+)
grade = 85
match grade:
    case g if g >= 90:
        print("A")
    case g if g >= 80:
        print("B")  # prints
    case g if g >= 70:
        print("C")
    case _:
        print("F")
```

```
B
```

Combining Operators in Real Checks

In [59]:
```python
# Relational + logical
age = 16
has_id = True
if (age >= 18) and has_id:
    print("Access granted")
else:
    print("Access denied")  # prints
```

```
Access denied
```

In [60]:
```python
# Membership
choice = "yes"
if choice.lower() in ["yes", "y"]:
    print("Proceed")  # prints
else:
    print("Stop")
```

```
Proceed
```

In [61]:
```python
# Identity vs equality
a = [1, 2, 3]
b = [1, 2, 3]
print(a == b)  # True: same values
print(a is b)  # False: different objects in memory
```

```
True
False
```

Quick Reminder: Control vs Sequential (with code)

In [62]:
```python
# Sequential structure (runs every line, top to bottom)
print("Start")          # 1. runs
total = 0               # 2. runs
total = total + 10      # 3. runs
print("Total:", total)  # 4. runs -> Total: 10
print("End")            # 5. runs
# Output always shows Start, Total: 10, End (no decisions made)
```

```
Start
Total: 10
End
```

In [63]:
```python
# Conditional control structure (decides which lines to run)
x = 7
print("Start")
if x > 10:
    print("x is greater than 10")
elif x > 5:
    print("x is greater than 5 but not more than 10")  # this runs
else:
    print("x is 5 or less")
print("End")
# Middle line changes depending on x
```

```
Start
x is greater than 5 but not more than 10
End
```

In [64]:
```python
count = 3
print("Countdown")
while count > 0:
    print(count)
    count = count - 1
print("Lift off!")
```

```
Countdown
3
2
1
Lift off!
```

Long if Control Strucutre Example

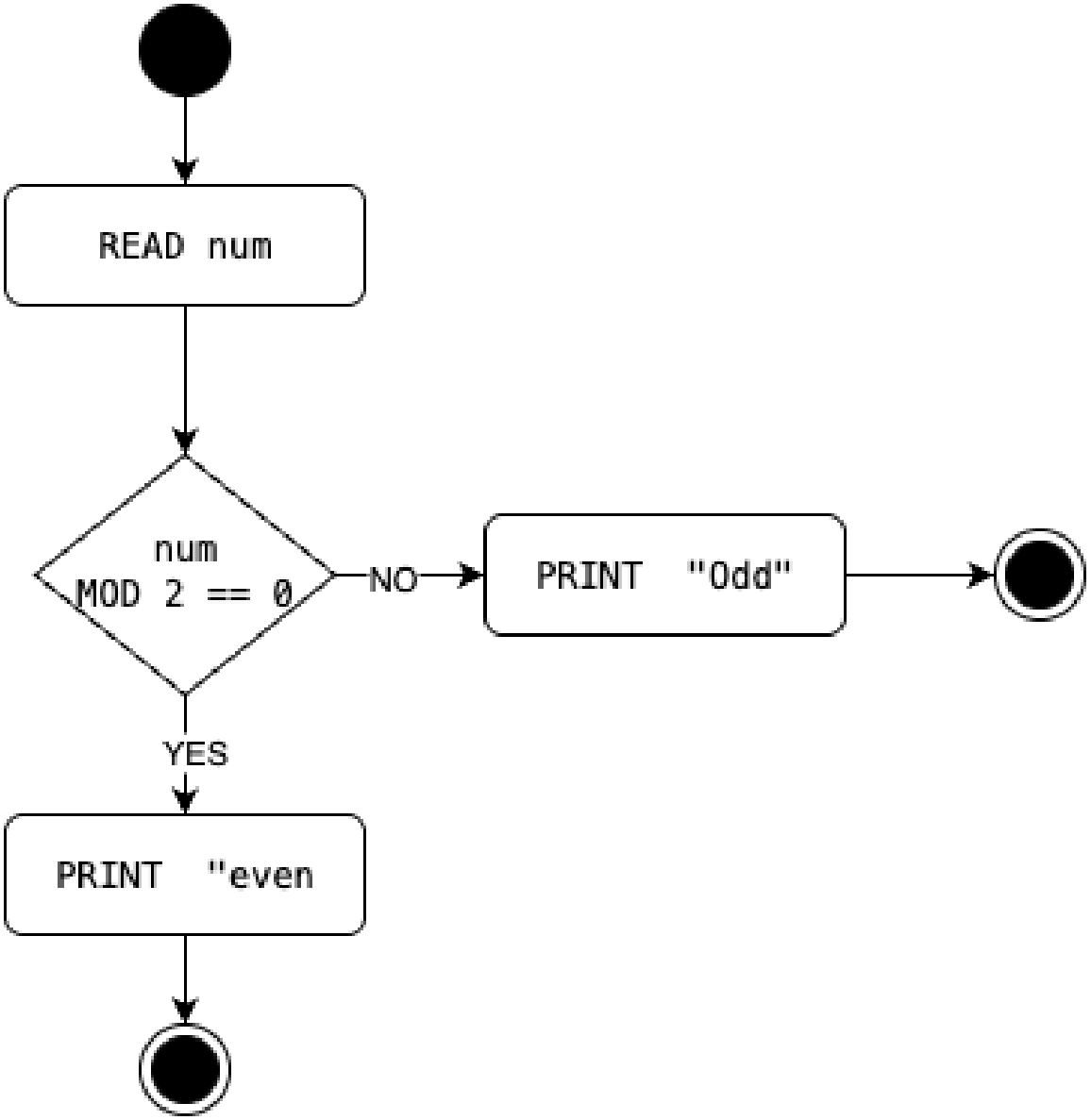
In [65]:
```python
age = 61
is_student = False
has_id = True
day = "wednesday"        # e.g., "monday"..."sunday"
time_24h = 16            # 0..23
```

```python
if (age >= 18 and has_id) or (age < 18 and has_id):
    # inside: we know has_id is True (any age), so check age buckets
    if age < 3:
        price = 0
        category = "infant (free)"
    elif age < 12:
        price = 60
        category = "child"
    elif age < 60:
        # adult: possible student discount
        if is_student and day in ["monday", "tuesday", "wednesday"]:
            price = 70
            category = "adult student midweek"
        else:
            price = 100
            category = "adult"
    else:
        # senior: nested time/day adjustments
        if day in ["monday", "tuesday", "wednesday", "thursday"] and time_24
            price = 60
            category = "senior matinee (weekday)"
        elif day in ["saturday", "sunday"]:
            # weekend surcharge then possible early-bird reduction
            if time_24h < 12:
                price = 70
                category = "senior early-bird (weekend)"
            else:
                price = 90
                category = "senior (weekend)"
        else:
            price = 75
            category = "senior (regular)"

    # global promos layered after category pricing
    # midweek promo for everyone at exactly 16:00 on Wednesday
    if day == "wednesday" and time_24h == 16:
        price = max(price - 10, 0)
        promo = "midweek-16 discount"
    else:
        promo = "none"

    print("access: granted")
    print("category:", category)
    print("promo:", promo)
    print("final price:", price)

else:
    # no id → no access
    print("access: denied - valid id required")
```

```
access: granted
category: senior matinee (weekday)
promo: midweek-16 discount
final price: 50
```

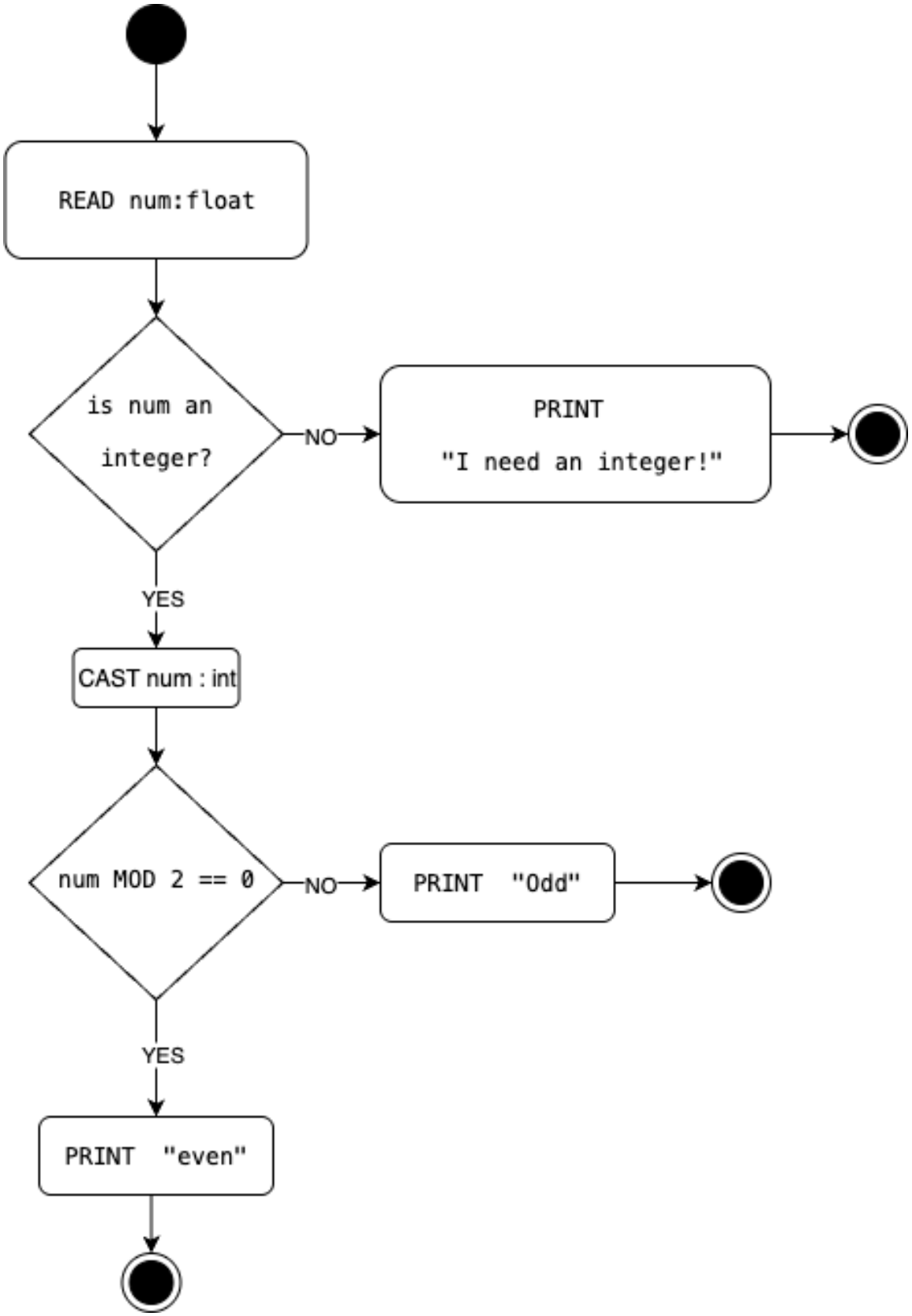
Exercises

1) Even / Odd



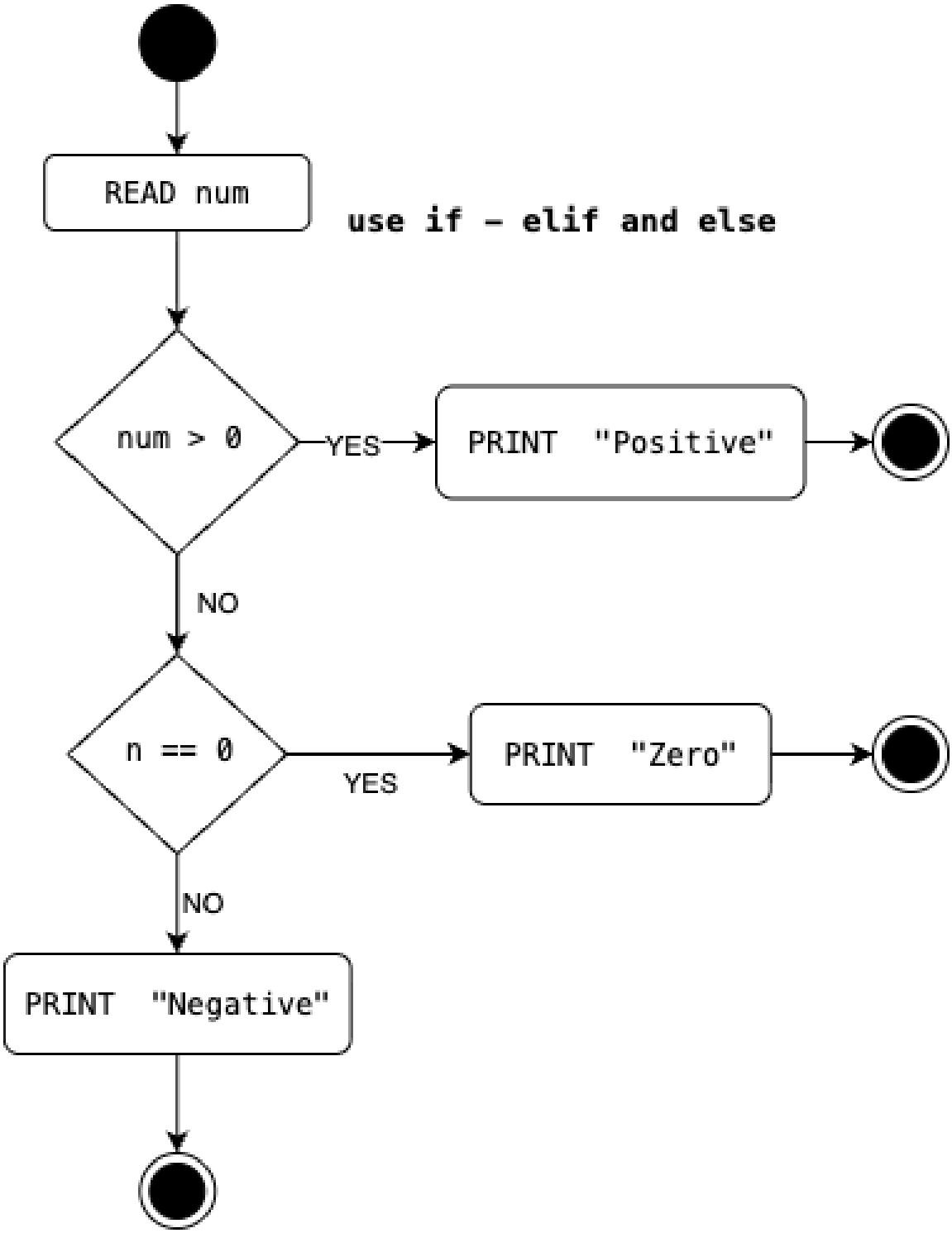
In [65]:

2) Even / Odd (integer check)



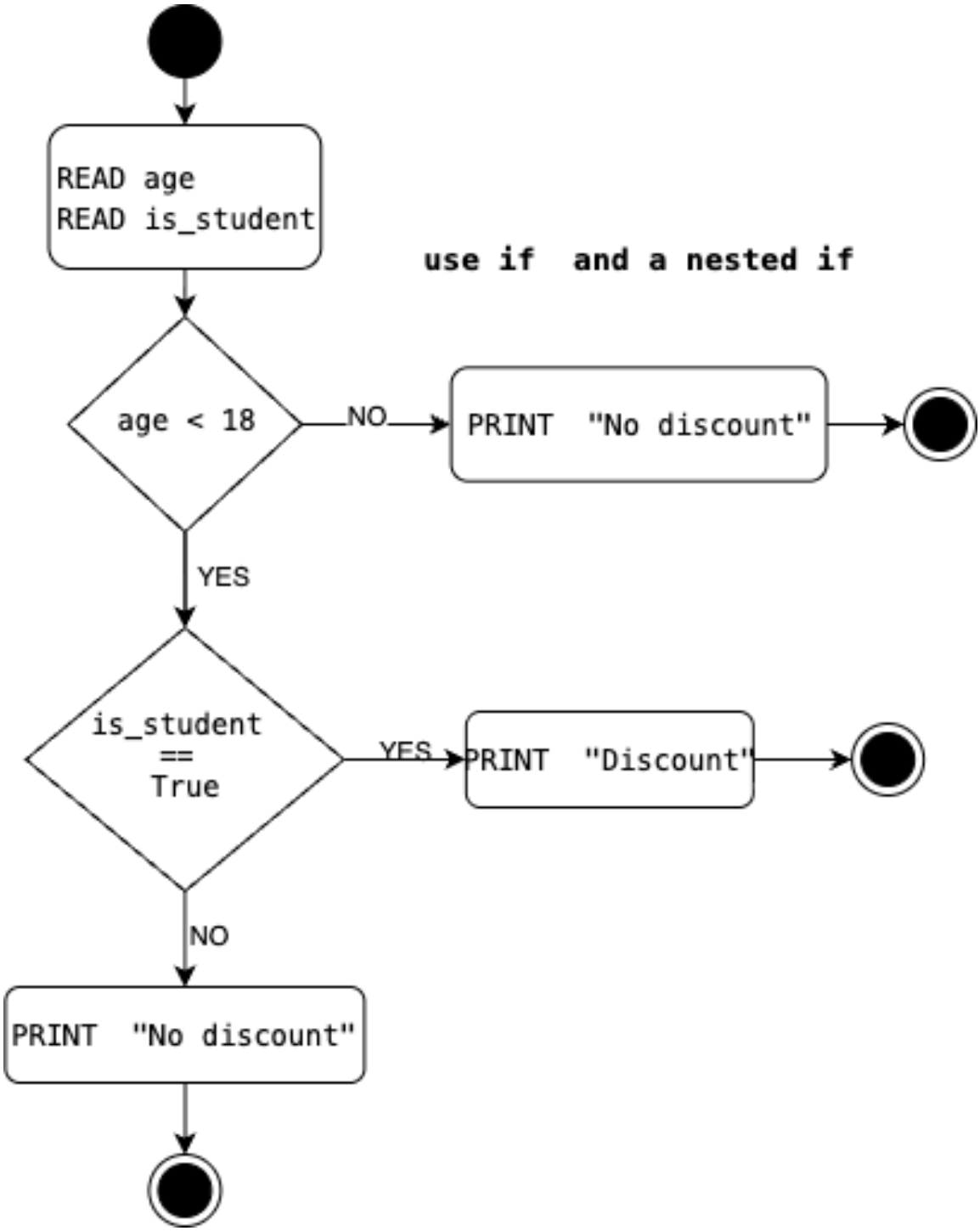
In [65]:

3) Positive / Negative / Zero (use if - elif and else)



In [65]:

4) Discount / No Discount (use if and a nested if)


READ age
READ is_student
use if and a nested if
age < 18
NO
PRINT "No discount"
YES
is_student
==
True
YES
PRINT "Discount"
NO
PRINT "No discount"

In [65]: