Structured programming

- Sequence structure
  - order to perform actions
- Selection structure (conditional, branches)
  - what to do depending on a decision
- Repetition structure (iteration, loops)
  - do something many times

Multiple line expressions

```
if ( condition ) {
    expression1
    expression2
    etc
}
```

This is a block of code

R

Multiple line expressions

```
satiation <- 42
if ( satiation < 50 ) {
    print("Squirrel is hungry")
    satiation <- satiation + 10
    print("Squirrel ate 10 nuts")
    print(paste("Satiation:", satiation))
}
```

Multiple line expressions

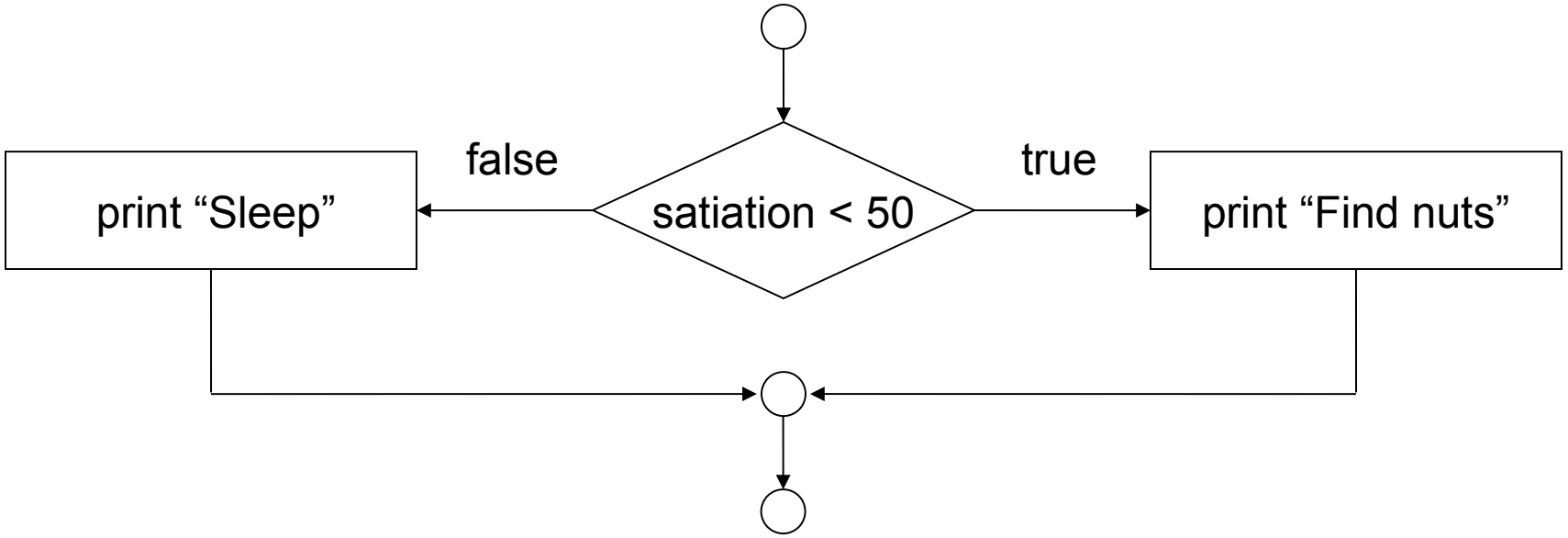
```
satiation = 42
if satiation < 50:
    print("Squirrel is hungry")
    satiation <- satiation + 10
    print("Squirrel ate 10 nuts")
    print("Satiation:", satiation)
```

Py

3 selection structures

if	single selection structure
if/else	double selection structure
if/else if	multiple selection structure

if/else selection structure



if/else selection structure

```
if ( condition ) {
     expression_1
} else {
     expression_2
}
```

all lines between braces indented 4 spaces

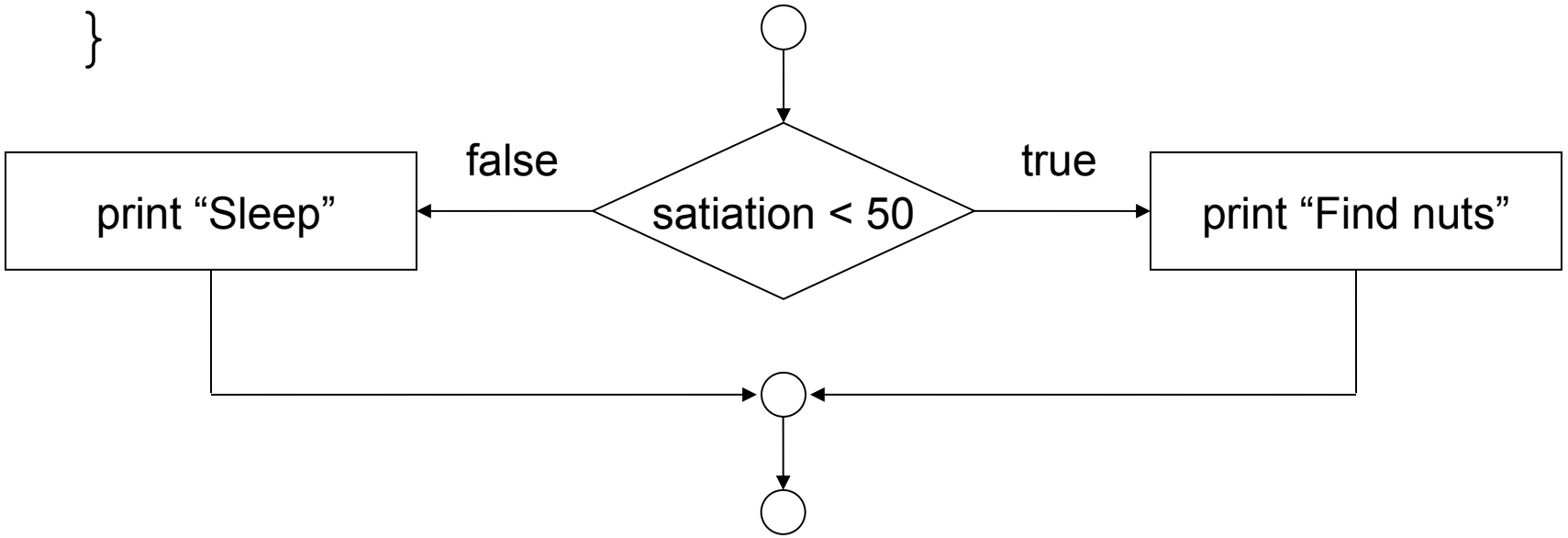
"} else" must be on same line

Good programming practice:
Always use braces, spacing, and indenting

if/else selection structure

```
if ( satiation < 50 ) {
     print("Find nuts")
} else {
     print("Sleep")
}
```

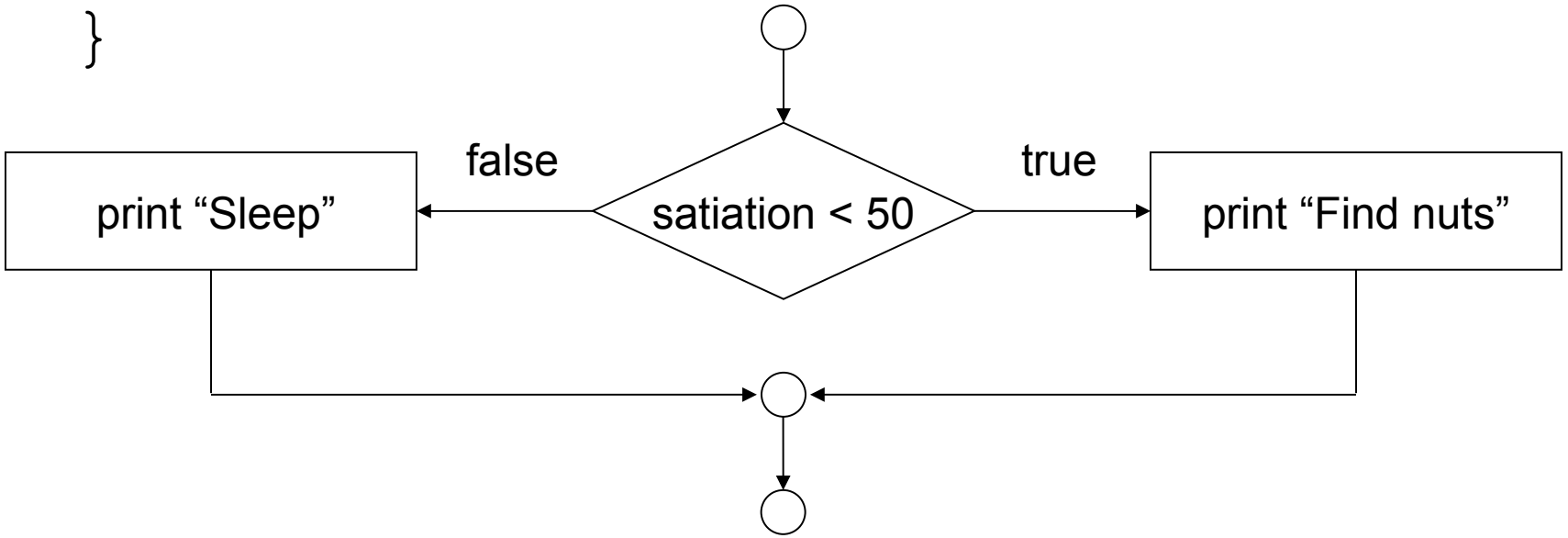
R



if/else selection structure

```
if ( satiation < 50 ) {
    print("Find nuts");
} else {
    print("Sleep");
}
```

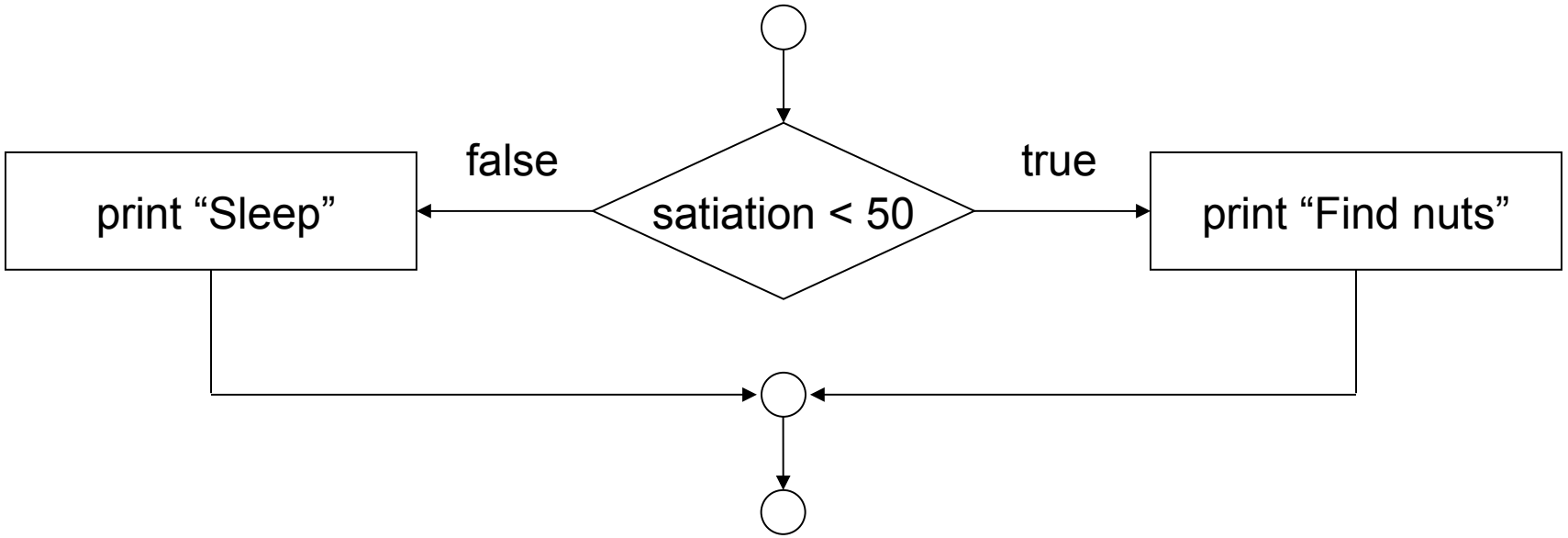
C



if/else selection structure

```
if satiation < 50:
    print("Find nuts")
else:
    print("Sleep")
```

Py



Combining control structures

- Stacking
  - one after another
- Nesting
  - one inside another

Stacked selection structures

```
soil_moisture <- 0.35
solar_radiation <- 2000
plant_stressed <- FALSE

if ( soil_moisture < 0.2 ) {
    plant_stressed <- TRUE
}
if ( solar_radiation > 1600 ) {
    plant_stressed <- TRUE
}
if ( plant_stressed ) {
    print("Plant is stressed")
}
```

R

Stacked selection structures

```
soil_moisture <- 0.35
solar_radiation <- 2000
plant_stress <- 0

if ( soil_moisture < 0.2 ) {
    plant_stress <- plant_stress + 10
}
if ( solar_radiation > 1600 ) {
    plant_stress <- plant_stress + 20
}
if ( plant_stress > 15 ) {
    print("Plant is stressed")
}
```

R

Nested selection structures

```
if ( density >= 70 ) {
     if ( density < 90 ) {
          color <- "Orange"
     }
}
```

What does this do?
Consider different values for density

R

Nested selection structures

- nested if/else structures
- creates an if/else if multiple selection structure

```
if ( cond1 ) {
     expression_1
} else {
     if ( cond2 ) {
          expression_2
     } else {
          expression_3
     }
}
```

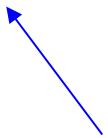
But don't write it this way.

R

Nested selection structures

- nested if/else structures
- creates an if/else if multiple selection structure

```
if ( cond1 ) {
    expression_1
} else if ( cond2 ) {
    expression_2
} else {
    expression_3
}
```



all lines between braces indented 4 spaces

R

Nested selection structures

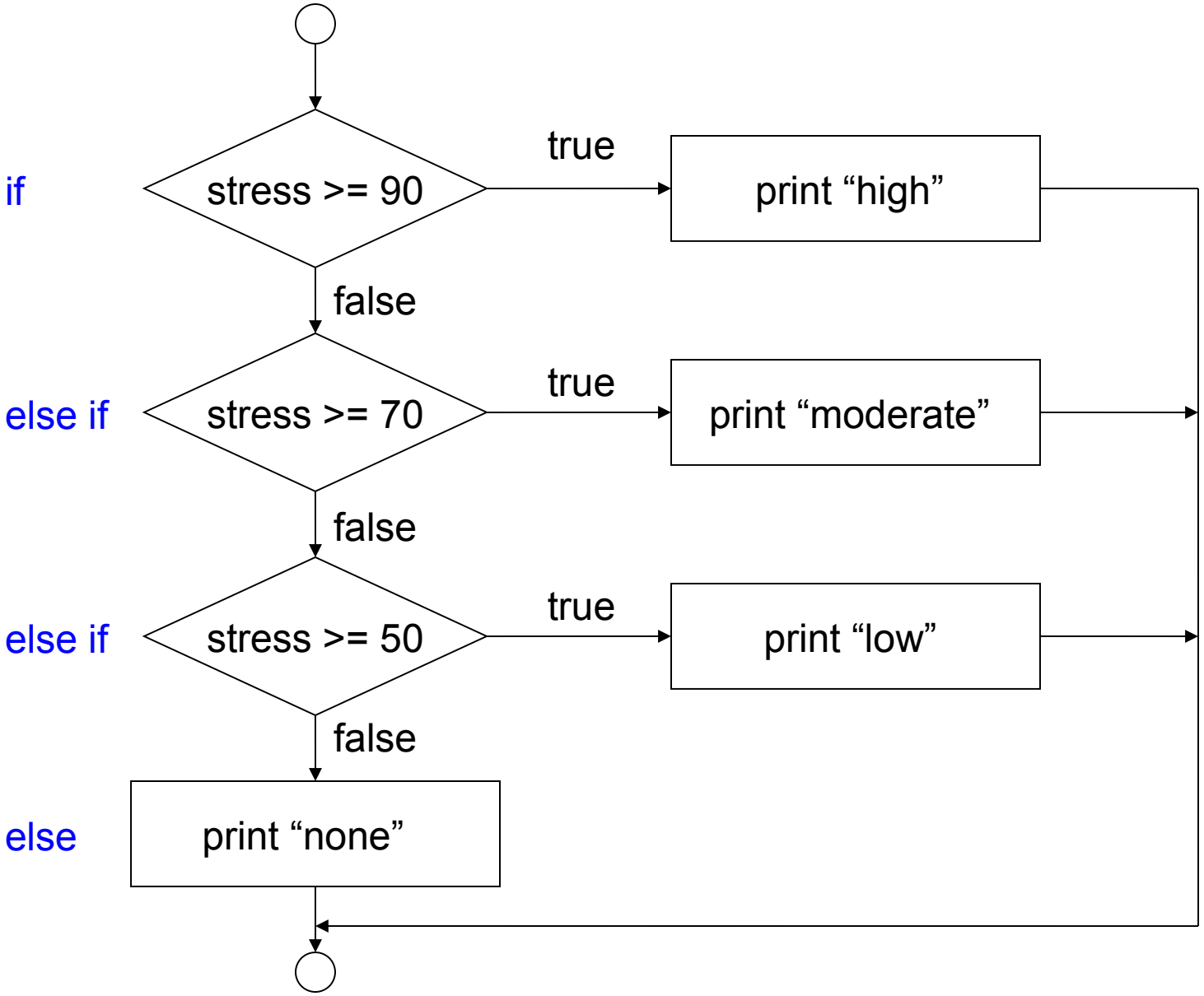
- nested if/else structures
- creates an if/else if multiple selection structure

```
if ( cond1 ) {
    expression_1;
} else if ( cond2 ) {
    expression_2;
} else {
    expression_3;
}
```

Nested selection structures

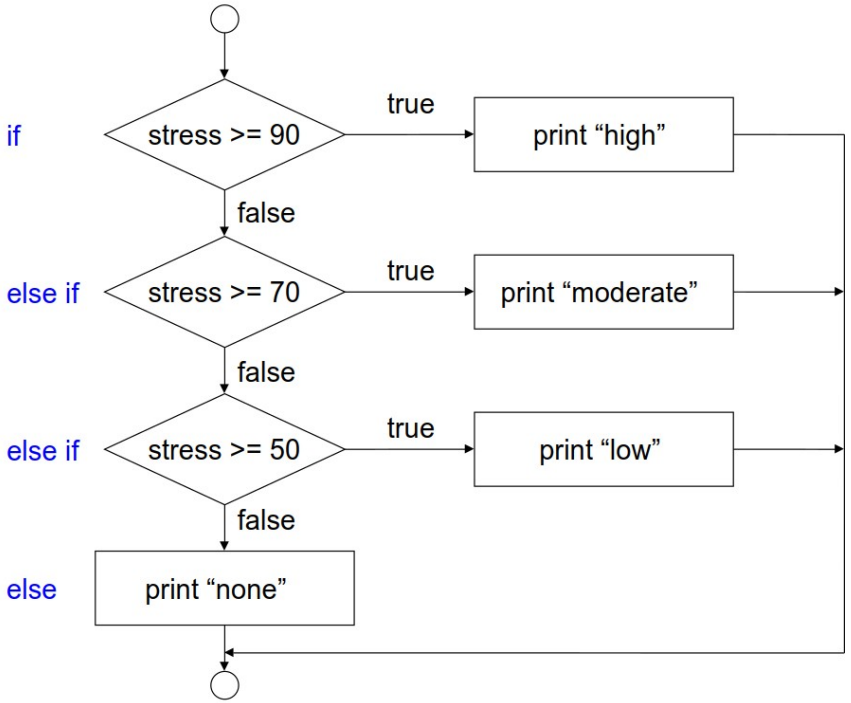
- nested if/else structures
- creates an if/else if multiple selection structure

```
if cond1:
    expression_1
elif cond2:
    expression_2
else:
    expression_3
```


if
stress >= 90
true
print “high”
false
else if
stress >= 70
true
print “moderate”
false
else if
stress >= 50
true
print “low”
false
else
print “none”

if/else if selection structure

```
if ( cond1 ) {
    expr1
} else if ( cond2 ) {
    expr2
} else if ( cond3 ) {
    expr3
} else {
    expr4
}
```



Exercise:
Code this in R and Python to print "Plant stress:", newline, "xxx", where "xxx" is “high”, “moderate”, “low”, or “none” depending on the plant stress level.
(run your code as a block by selecting it all and hitting ctrl-return)

if is fundamental

```
hungry = satiation < 50
if hungry:                          if
    print("Find nuts")
if !hungry:                         else
    print("Sleep")
```

This shows that an if/else structure can be built from the fundamental if structure. We would not do this in practice. We would use an if/else structure instead.

Py