technoteach

technocamps

Llywodraeth Cymru
Welsh Government

Prifysgol Abertawe
Swansea University

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYD

PRIFYSGOL
BANGOR
UNIVERSITY

Cardiff Metropolitan University
Prifysgol Metropolitan Caerdydd

University of South Wales
Prifysgol De Cymru

Cyngor Cyllido Addysg Uwch Cymru
Higher Education Funding Council for Wales
hefcw

Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

PRIFYSGOL
ABERYSTWYTH
UNIVERSITY

PRIFYSGOL
Glyndŵr
Wrecsam
Wrexham
Glyndŵr
UNIVERSITY

institute of
CODING
in wales
technocamps

Programming Principles

Programming Principles

Good programming requires adherence to a set of principles

Your learners should possess an understanding of the following principles:

- Readability
- Maintainability
- Extensibility
- Modularity

These principles often feed into each other

technoteach technocamps

Readability

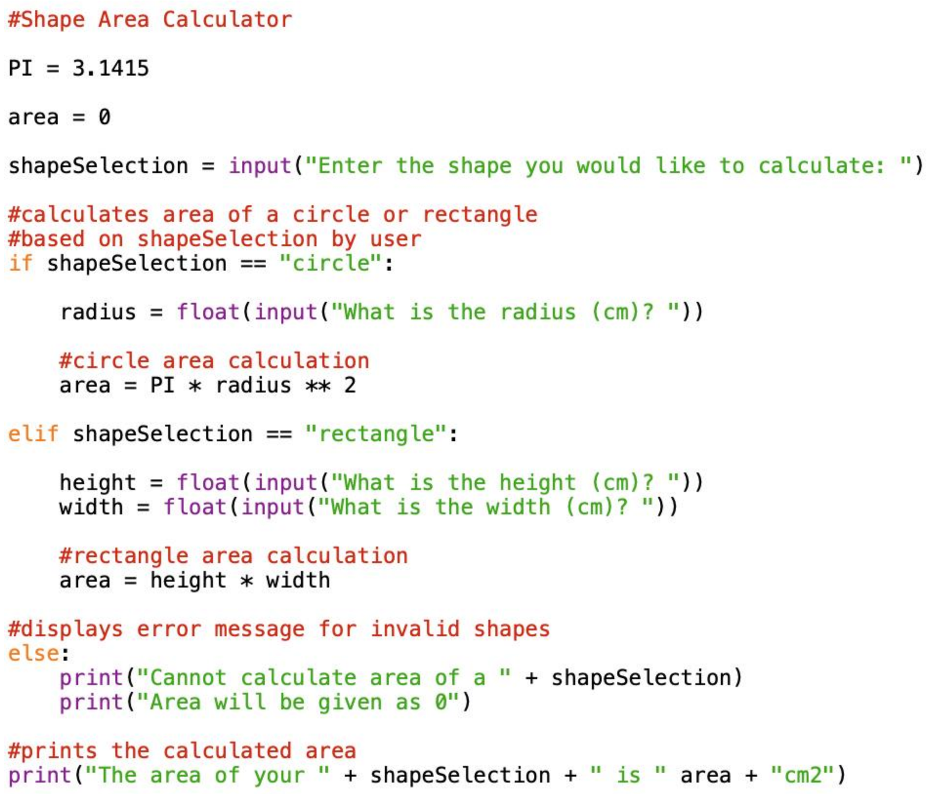
How readable the code is:

- What is the purpose?
- How does it work?

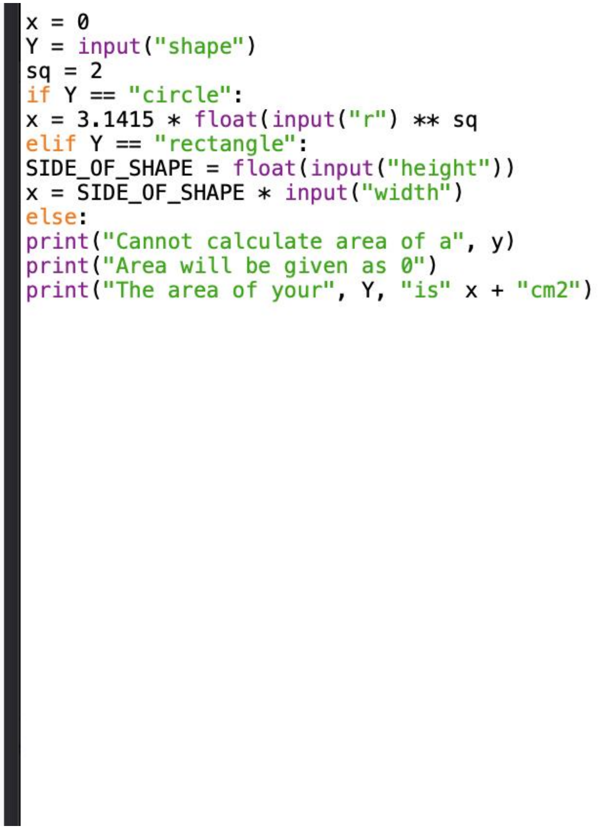
Many factors contribute to readability:

- Indentation
- Commenting
- Naming conventions
- Decomposition (links to modularity)
- Style consistency

Readability



```
#Shape Area Calculator

PI = 3.1415

area = 0

shapeSelection = input("Enter the shape you would like to calculate: ")

#calculates area of a circle or rectangle
#based on shapeSelection by user
if shapeSelection == "circle":

    radius = float(input("What is the radius (cm)? "))

    #circle area calculation
    area = PI * radius ** 2

elif shapeSelection == "rectangle":

    height = float(input("What is the height (cm)? "))
    width = float(input("What is the width (cm)? "))

    #rectangle area calculation
    area = height * width

#displays error message for invalid shapes
else:
    print("Cannot calculate area of a " + shapeSelection)
    print("Area will be given as 0")

#prints the calculated area
print("The area of your " + shapeSelection + " is " area + "cm2")
```

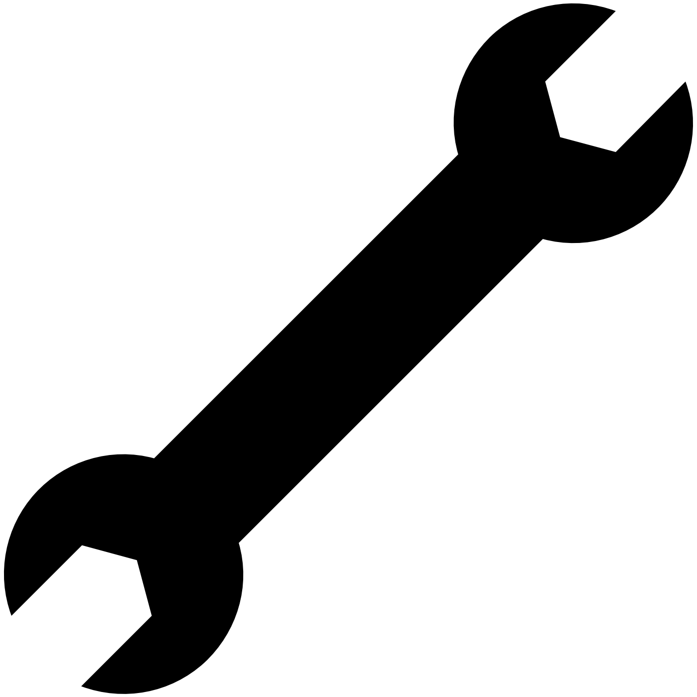


```
x = 0
Y = input("shape")
sq = 2
if Y == "circle":
x = 3.1415 * float(input("r") ** sq
elif Y == "rectangle":
SIDE_OF_SHAPE = float(input("height"))
x = SIDE_OF_SHAPE * input("width")
else:
print("Cannot calculate area of a", y)
print("Area will be given as 0")
print("The area of your", Y, "is" x + "cm2")
```

Which of these is more readable?

Shorter code != readability

Maintainability



The ease with which a program can be modified

Used to:

- Improve & customise code
- Bug fixing
- Toughening security
- Port the program to new software environments

The outcome of high readability, extensibility and modularity

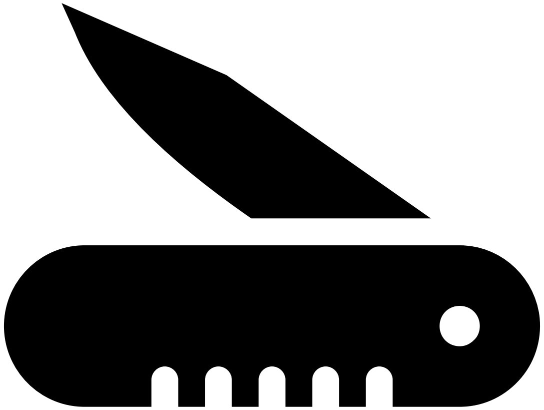
Extensibility

The ease with which new behaviour may be implemented within an existing program

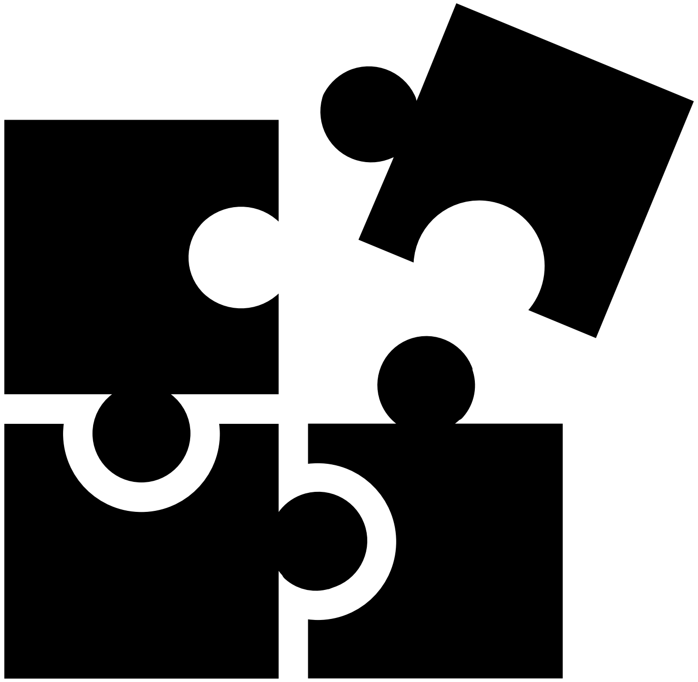
Covers the addition of entirely new functionality as well as the expansion of existing functions

Enables the expansion of program capabilities without impacting the original functionality

Closely linked to readability and modularity



Modularity



The breaking down of complex code into individual modules

Allows for work to be split amongst multiple developers

Modules may be individually written and tested before integration

Good modularity requires system modules to be designed with high cohesion and low coupling

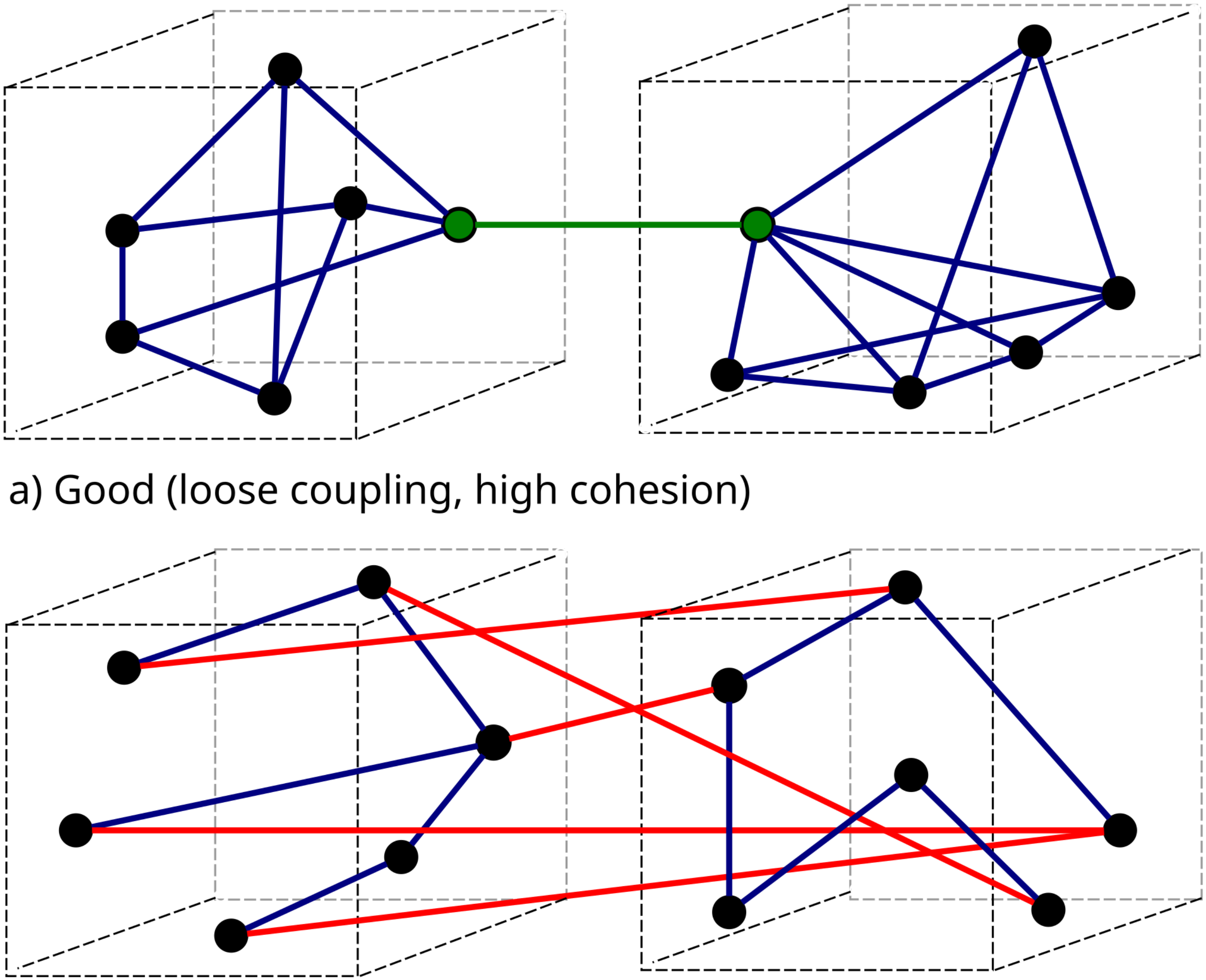
Cohesion & Coupling

Cohesion

- How much the elements of a single module belong together
- Are all variables, methods etc. closely linked?
- Are all elements relevant to meeting stated purpose of the module?

Coupling

- How much the modules of a system depend on each other
- How closely connected are different modules to each other?
- How strong are those relationships between the modules - does one require another to function?



a) Good (loose coupling, high cohesion)

b) Bad (high coupling, low cohesion)

The links between module elements in systems with good (top) and bad (bottom) cohesion and coupling