# Top Down Design

## Question: 1

Top Down Design makes it easier to solve a problem by:

Breaking the problem down into smaller parts

Starting your code from the first command

Starting with the biggest function and then moving to smaller ones

Making each function hold only one command

## Question: 2

Which of the following is NOT an example of using Top Down Design?

Breaking a large code up into functions

Writing code for one part of a function before tackling the whole thing

Using descriptive names for variables

Separating a large problems into smaller ones

# Variables

## Question: 1

Variables allow us to:

Name different parts of our programs

Use english words to communicate with Tracy

Store information to use in our programs

Change the words Tracy recognizes

## Question: 2

Which program will have Tracy move forward 10, then turn left and move forward 20?

distance = 10

forward(distance)

left(90)

distance = distance \* 2

forward(distance)

distance = 10

forward(distance)

left(90)

forward(distance)

distance = 10

distance = distance \* 2

forward(distance)

left(90)

forward(distance)

distance = 10

forward(distance)

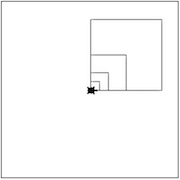
left(90)

forward(distance)

distance = distance \* 2

## Question: 3

Which of the following programs would produce this output?



square\_length = 20

for i in range(4):

forward(square\_length)

left(90)

square\_length = square\_length \* 2

square\_length = 20

for i in range(4):

for i in range(4):

forward(square\_length)

left(90)

square\_length = square\_length \* 2

square\_length = 20

for i in range(4):

forward(square\_length)

left(90)

square\_length = square\_length \* 2

square\_length = 20

for i in range(4):

for i in range(4):

forward(square\_length)

left(90)

square\_length = square\_length \* 2