



Through the process of **ABSTRACTION**, a programmer hides all but the relevant data about an object in order to reduce complexity and increase efficiency.

Another definition: when you ABSTRACT, you make the code less about the specific case and more about the general case.

In the same way that ABSTRACTION is used in art, the object that remains is a representation of the original with unwanted detail omitted.

# CODING FOR SPECIFIC CASE

```
when 5 key pressed
repeat 4
  if item 1 of My List > item 2 of My List then
    insert item 2 of My List at 1 of My List
    delete 3 of My List
  if item 2 of My List > item 3 of My List then
    insert item 3 of My List at 2 of My List
    delete 4 of My List
  if item 3 of My List > item 4 of My List then
    insert item 4 of My List at 3 of My List
    delete 5 of My List
  if item 4 of My List > item 5 of My List then
    insert item 5 of My List at 4 of My List
    delete 6 of My List
```

The code on the left is a Scratch script that implements a specific case of a sorting algorithm. It starts with a 'when 5 key pressed' block, followed by a 'repeat 4' loop. Inside the loop, there are four 'if' blocks. Each 'if' block checks if the current item is greater than the next item. If true, it inserts the next item at the current position and deletes the current item. This process repeats for the first four items of the list, effectively sorting the first five items.

# BEGINNING TO ABSTRACT

```
when 5 key pressed
repeat 4
  set n to 1
  if item n of My List > item n + 1 of My List then
    insert item n + 1 of My List at n of My List
    delete n + 2 of My List
  change n by 1
  if item n of My List > item n + 1 of My List then
    insert item n + 1 of My List at n of My List
    delete n + 2 of My List
  change n by 1
  if item n of My List > item n + 1 of My List then
    insert item n + 1 of My List at n of My List
    delete n + 2 of My List
  change n by 1
  if item n of My List > item n + 1 of My List then
    insert item n + 1 of My List at n of My List
    delete n + 2 of My List
```

The code on the right is a Scratch script that implements an abstract version of the same sorting algorithm. It starts with a 'when 5 key pressed' block, followed by a 'repeat 4' loop. Inside the loop, there are four iterations. Each iteration starts with a 'set n to 1' block (highlighted in yellow), followed by an 'if' block that checks if the current item is greater than the next item. If true, it inserts the next item at the current position and deletes the current item. This process repeats for the first four items of the list, effectively sorting the first five items. The variable 'n' is updated to 'n + 1' at the end of each iteration.

# BEGINNING TO ABSTRACT (from previous)

# ABSTRACTED

```
when s key pressed
repeat 4
  set n to 1
  if item n of My List > item n + 1 of My List then
    insert item n + 1 of My List at n of My List
    delete n + 2 of My List
  change n by 1
  if item n of My List > item n + 1 of My List then
    insert item n + 1 of My List at n of My List
    delete n + 2 of My List
  change n by 1
  if item n of My List > item n + 1 of My List then
    insert item n + 1 of My List at n of My List
    delete n + 2 of My List
  change n by 1
  if item n of My List > item n + 1 of My List then
    insert item n + 1 of My List at n of My List
    delete n + 2 of My List
```

The code on the left is a linear sequence of operations. It starts with a 'when s key pressed' block, followed by a 'repeat 4' block. Inside this repeat block, there are four identical sets of code, each corresponding to a value of  $n$  (1, 2, 3, and 4). Each set includes an 'if' statement that checks if the item at index  $n$  is greater than the item at index  $n + 1$ . If true, it inserts the item at  $n + 1$  at index  $n$  and then deletes the item at index  $n + 2$ . After each 'if' block, the value of  $n$  is incremented by 1. The code is annotated with yellow boxes and labels: 'n=1', 'n=2', 'n=3', and 'n=4' are placed next to the 'change n by 1' blocks. A yellow border highlights the entire 'repeat 4' block.

```
when s key pressed
repeat 4
  set n to 1
  repeat 4
    if item n of My List > item n + 1 of My List then
      insert item n + 1 of My List at n of My List
      delete n + 2 of My List
    change n by 1
```

The code on the right is an abstracted version of the code on the left. It starts with a 'when s key pressed' block, followed by a 'repeat 4' block. Inside this repeat block, there is a 'set n to 1' block, followed by another 'repeat 4' block. This inner repeat block contains an 'if' statement that checks if the item at index  $n$  is greater than the item at index  $n + 1$ . If true, it inserts the item at  $n + 1$  at index  $n$  and then deletes the item at index  $n + 2$ . After each 'if' block, the value of  $n$  is incremented by 1. The code is annotated with a yellow box highlighting the inner 'repeat 4' block, indicating that the four iterations of the original code are now represented by a single 'repeat 4' block.