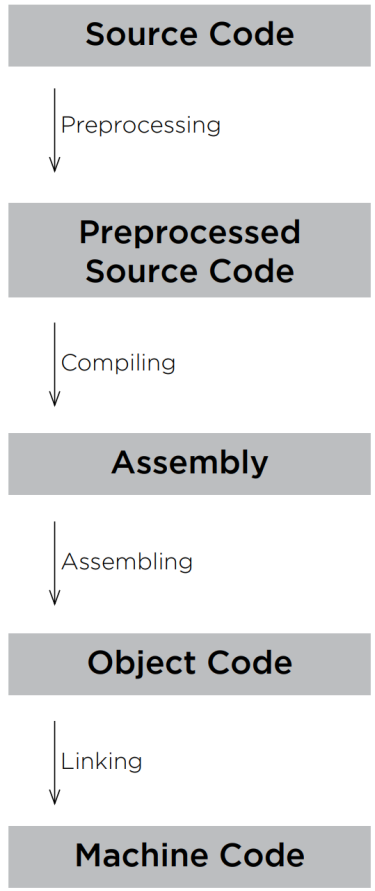
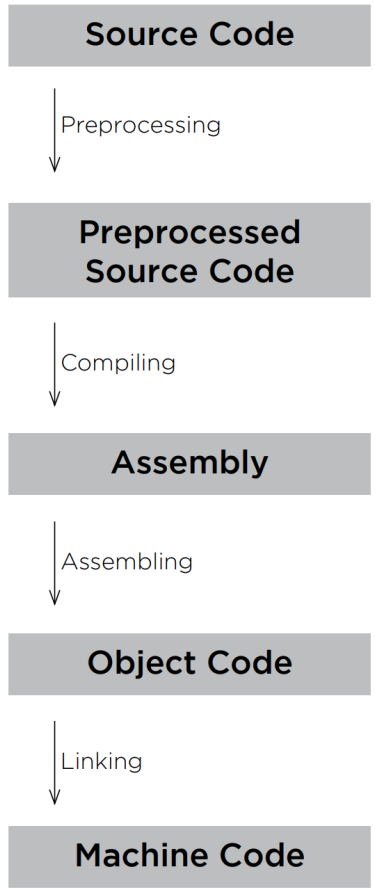
Chapter 2 Review



Compiling

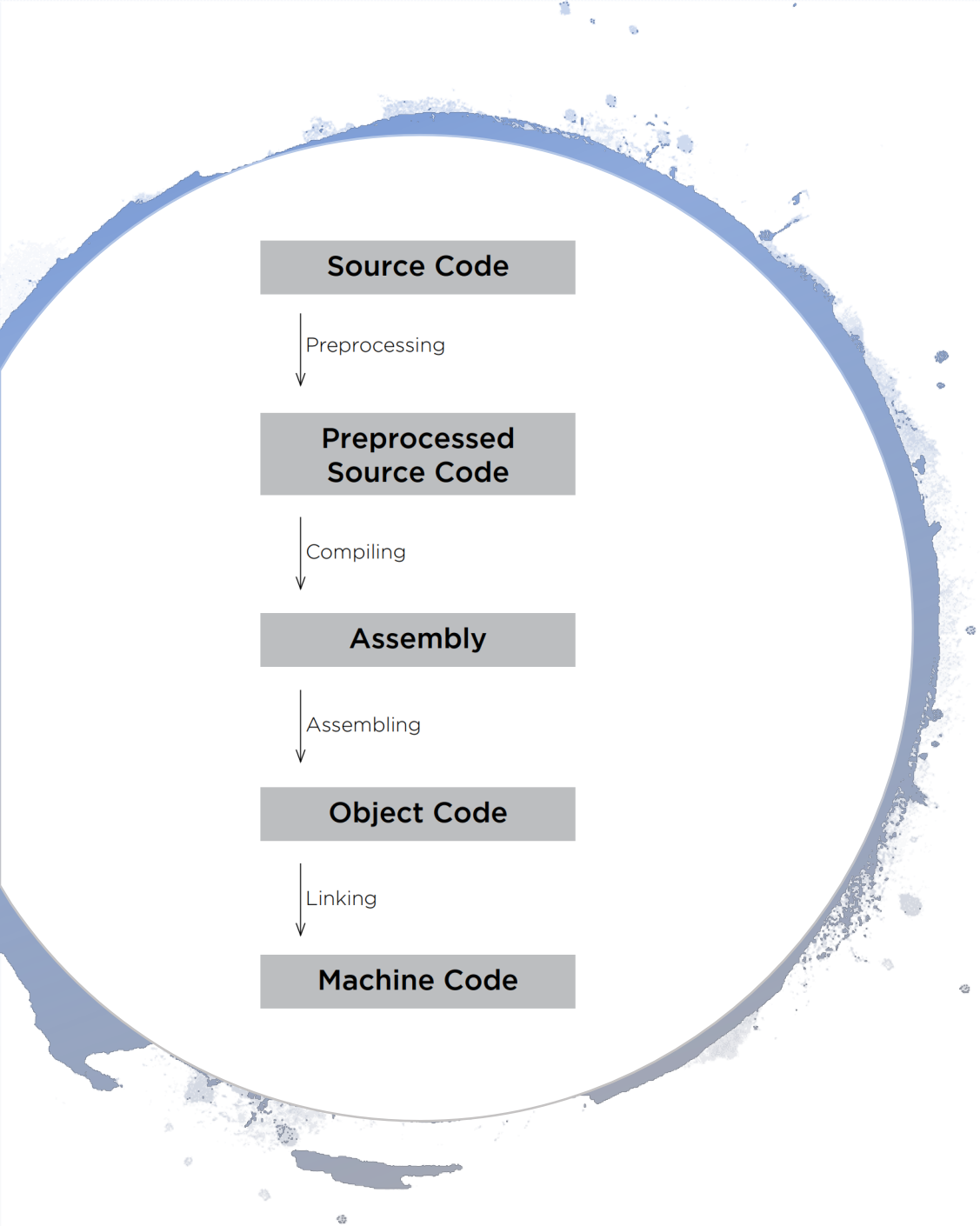
- Preprocessing
- Compiling
- Assembling
- Linking



Compiling

Preprocessing

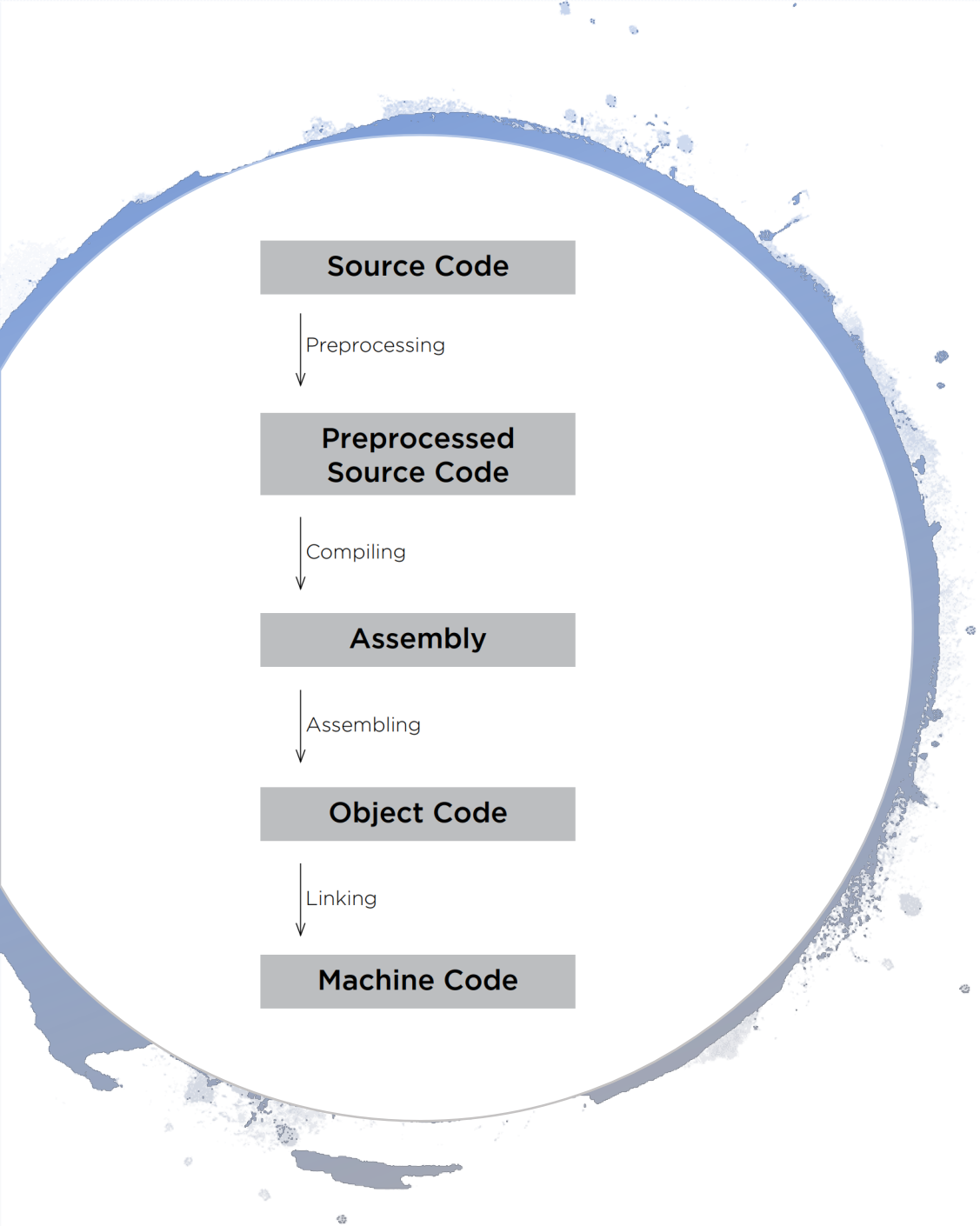
- Takes the #includes at the start of our program, and tells the preprocessor to include the contents of that file(s) in our preprocessed file
- This replaces the the #include with the ENTIRE contents of the .h file



Compiling

Compiling

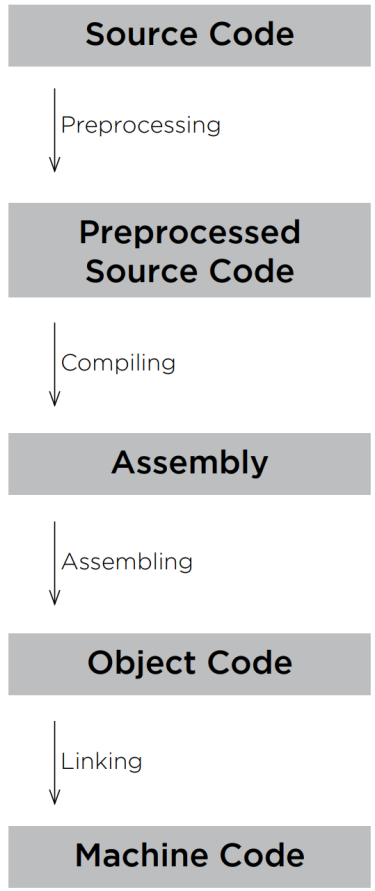
- Takes the preprocessed source code, and compiles it into lower-level language called **assembly**
- This is much closer to what the computer can understand, while still being somewhat readable for humans



Compiling

Assembling

- Turns the assembly code into **object code**
- This is essential machine code with some non-machine code symbols
- If there are no other files that need to be combined with the source code, compiling is over at this point



Compiling

Linking

- This final step takes multiple files (like math.h or cs50.h) and combines them with our object code to create one singe file of machine code

Arrays & Strings

- Stores multiple values of the same data type
- Arrays start at index ____
- A string is simply an array of _____
- What is the null terminator?

Typecasting

Explicit Typecasting

```
printf("%c\n", (char) x);
```

Implicit Typecasting

```
printf("%c\n", x);
```

Used when you need to convert one data type to another

Only constraint is that you cannot go from _____ to _____ precise

Command-Line Interaction

- Key Terms:
  - Command-line arguments
  - Argument count – What is the data type?
  - Argument Vector – What is this data type?

Exit Codes

Default return value of main is ____

Input validation

Debugging

Searching and Sorting

Algorithm Name	Basic Concept	Worst-case runtime (O)	Best-case runtime (Ω)
Bubble Sort	Swap **adjacent pairs** of elements if they are out of order, effectively "bubbling" larger elements to the right and smaller ones to the left.	n^2	n
Selection Sort	Find the **smallest** unsorted element in an array and swap it with the first unsorted element of that array.	n^2	n^2
Insertion Sort	Proceed once through the array from left -to-right, **shifting** elements as necessary to insert each element into its correct place	n^2	n
Merge Sort	Split the fill array into subarrays, then merge those subarrays back together in the correct order	n log n	n log n
Linear Search	Search the array from left to right in order until the target value is found, or the entire array has been searched	n	1
Binary Search	Find the middle point of the array, if it is the target, stop, otherwise see if target is less than or greater than, and repeat	log n	1

Recursion

- **Recursive solutions** implement a function that repeatedly invokes a slightly modified instance of itself with each subsequent instance tending closer and closer to a **base case**
- These intermediate calls are left waiting for the base case to start passing everything back up
- What would happen if we didn’t have a base case?