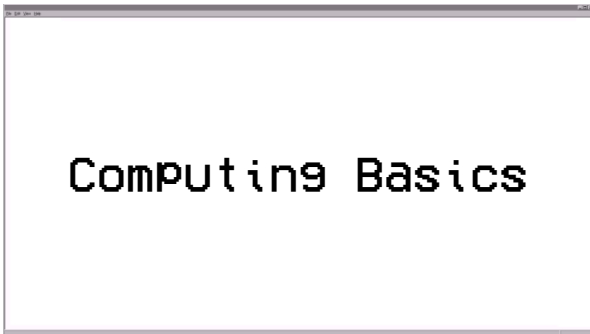




1



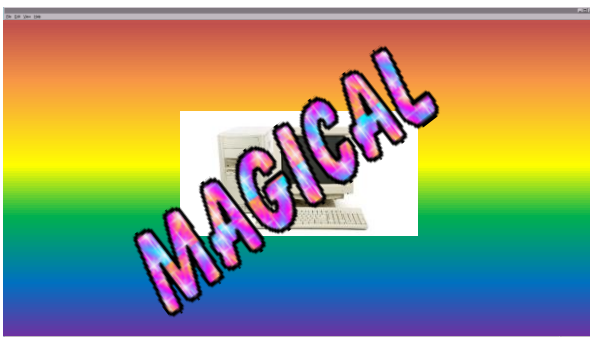
2



3



4



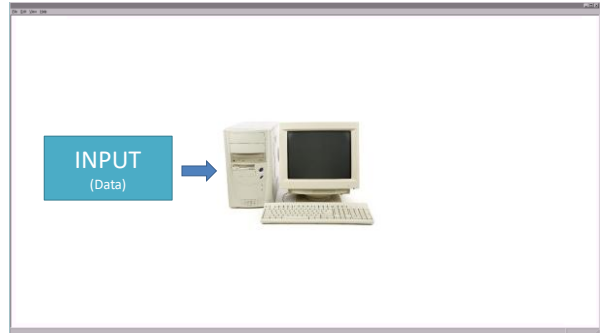
5



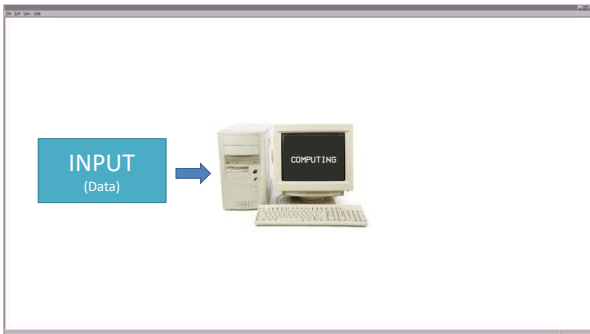
6



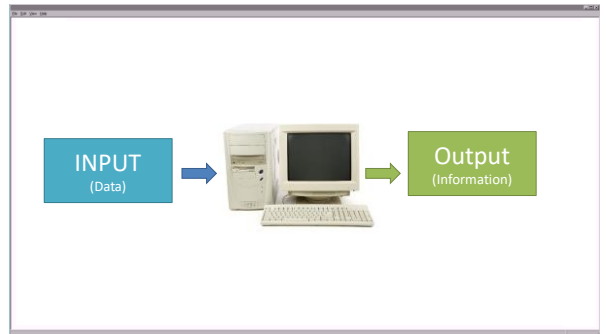
7



8



9



10



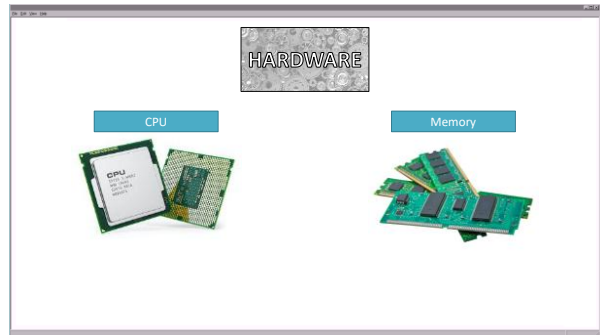
11



12



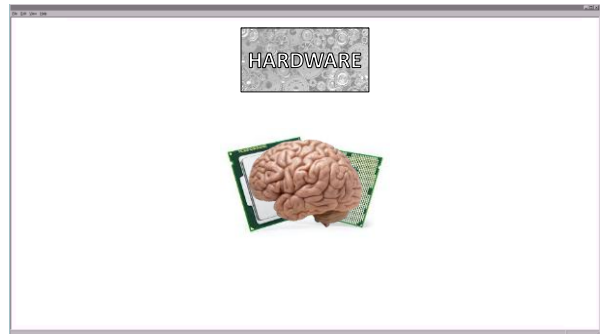
13



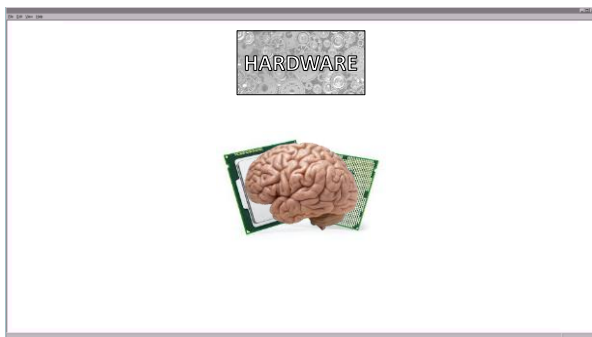
14



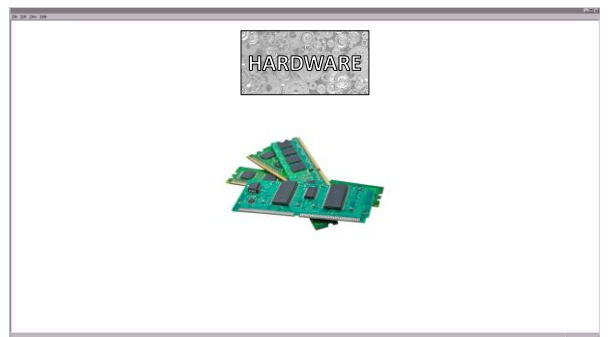
15



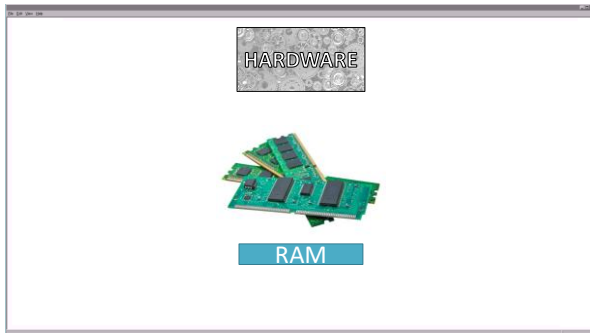
16



17



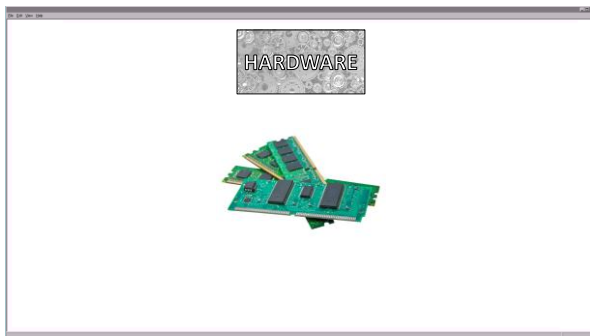
18



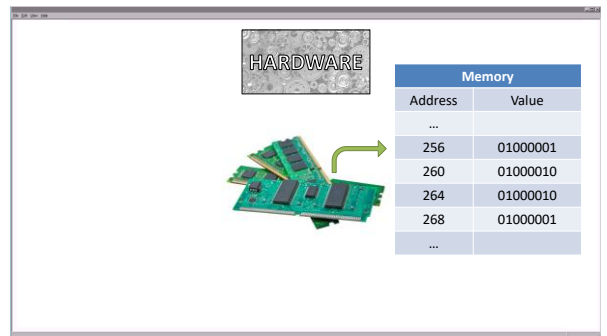
19



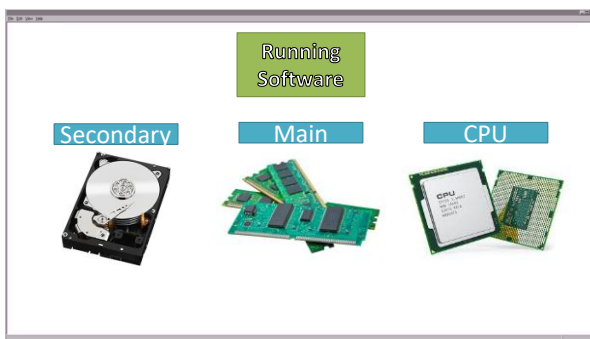
20



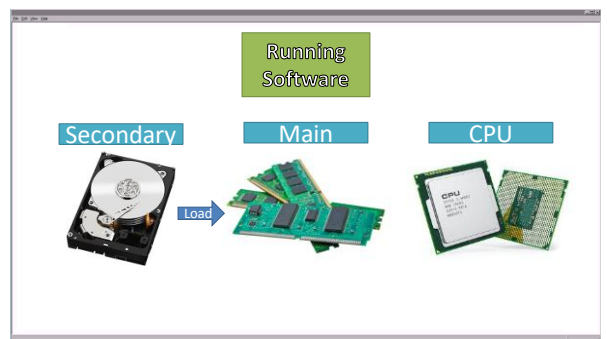
21



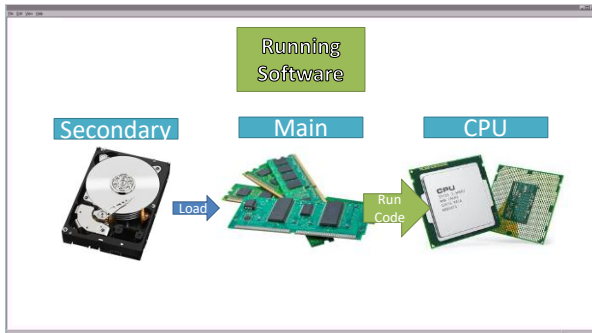
22



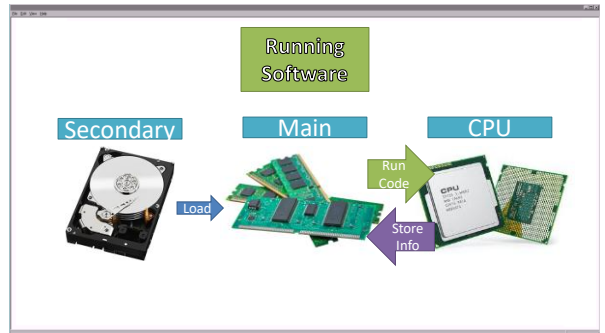
23



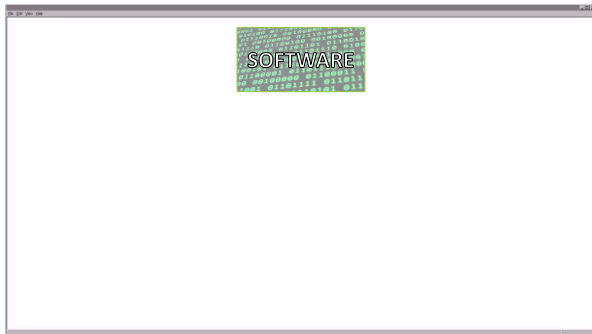
24



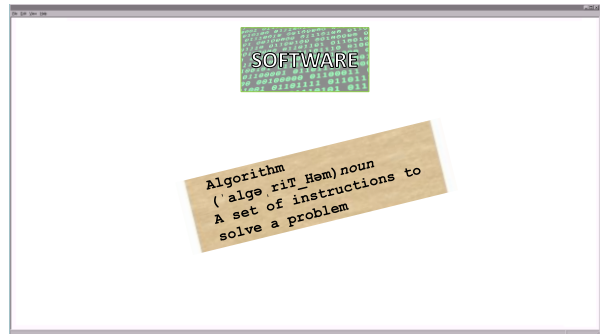
25



26



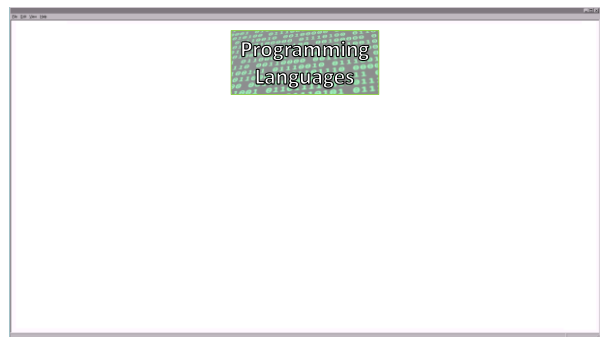
27



28



29



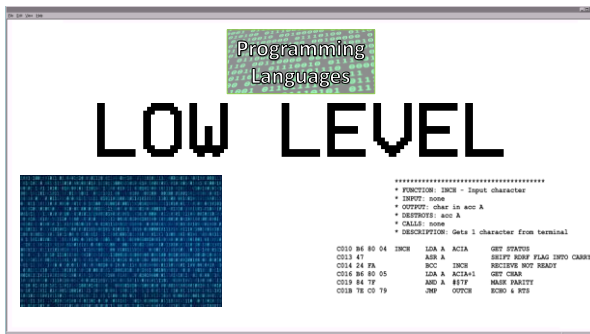
30



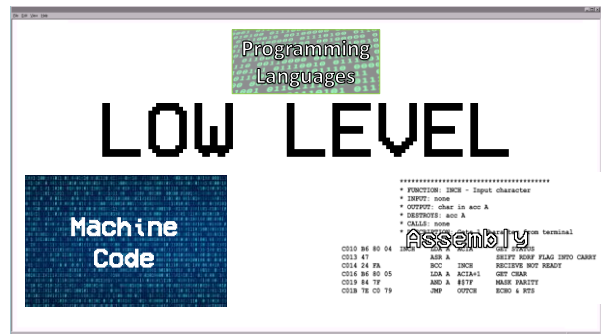
31



32



33



34



35



36



37



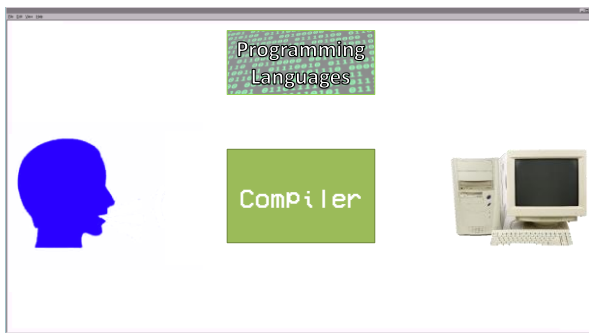
38



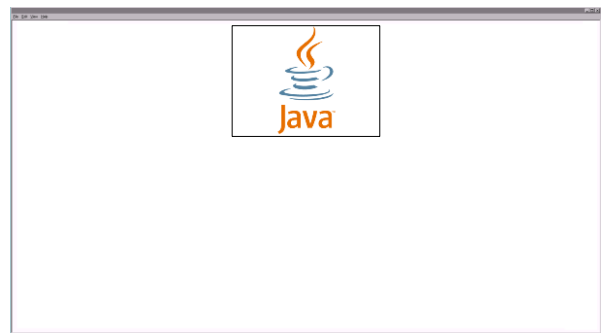
39



40



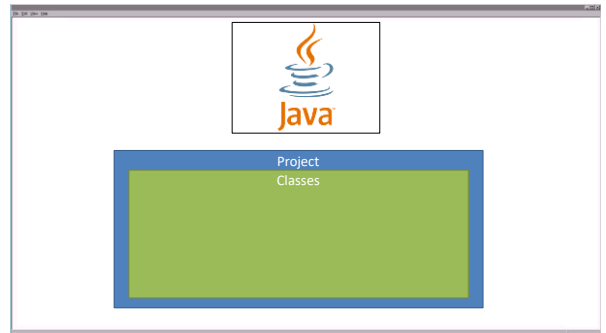
41



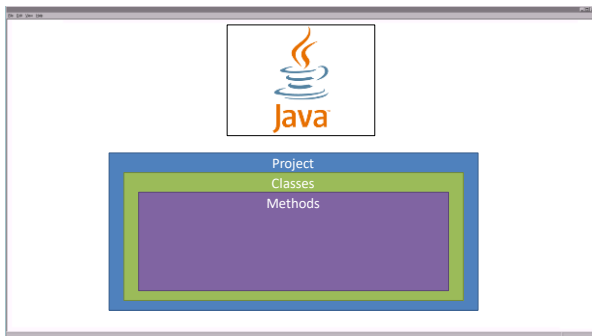
42



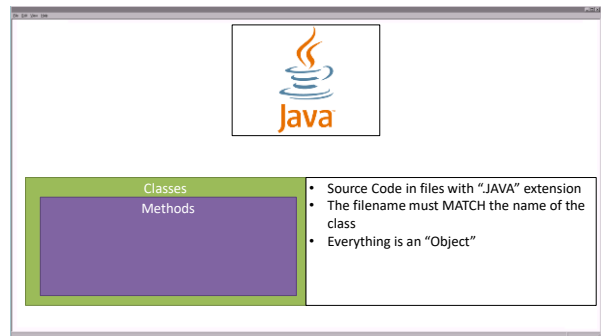
43



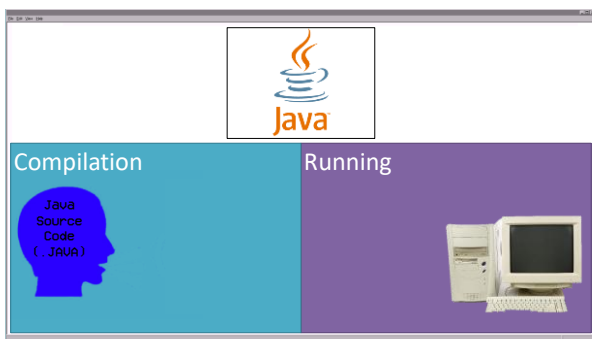
44



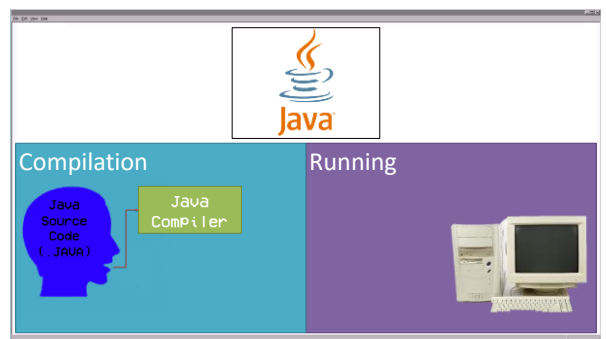
45



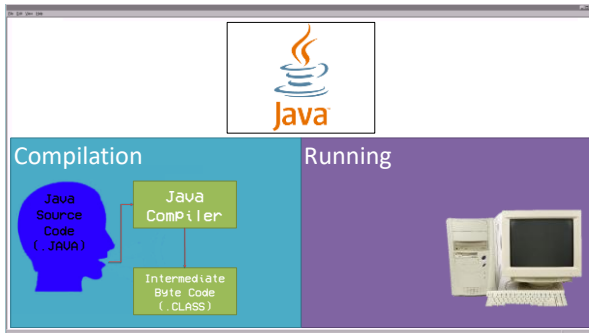
46



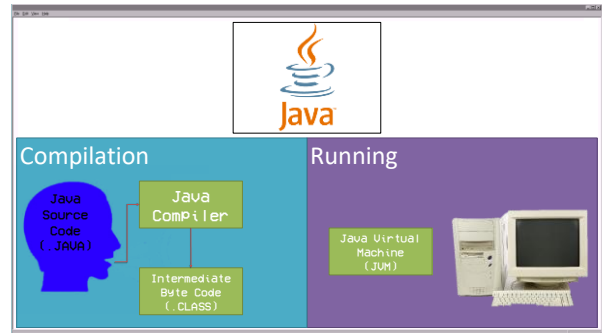
47



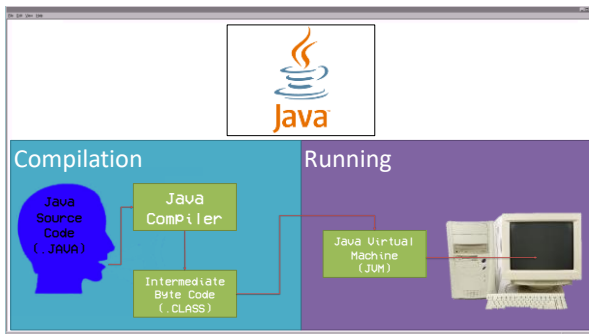
48



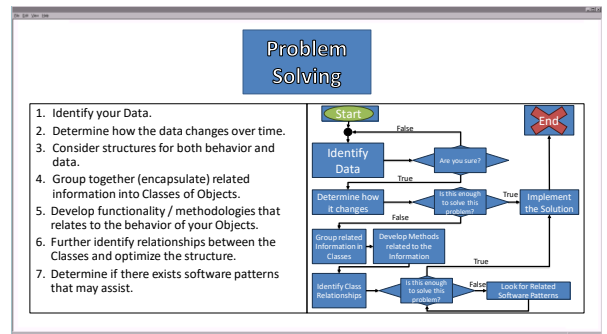
49



50



51



52

HELLO WORLD!

53

HELLO **WORLD!**

54

HELLO ARRAYS!

Arrays

- A collection (data structure) of items of the same type
- Fixed, contiguous block in memory
- Cannot resize in memory
 - Size of the array needs to be known before it is created
- Java arrays are considered "Objects"
 - Separated reference and contents
 - Have built in properties like "length"
- When arrays are constructed all items are assumed to be assigned default values, in Java
- Indices (singular "index") is how we can access and modify values in an array
- Valid indices start from 0 until Length - 1
 - If an array had 10 elements, then the valid indices are from 0 to 9
- Array's "best friend" is a for-loop
 - The loop can index into the array using its counter

Syntax

```
//Declaring and Constructing an Array
<<type>[] <identifier> = new <<type>[<size>];
//Indexing into an array to access a value
<identifier>[<index>];
//Indexing into an array to assign / modify a value
<identifier>[<index>] = <value>;
```

Examples

```
int[] i = new int[5];
i[2] = 22;
System.out.println(i[2]);
```

Sorting Algorithms

- **Problem:**
– Given any array of integers, develop an algorithm that sorts the values from smallest to largest.
- **Selection Sort**
1. Start from index 0
 2. Assume the starting index has the smallest value and record that index
 3. Sequentially check every other value
 4. If a value is found that is smaller at another index, then record that current index
 5. Once all values have been checked if the recorded index does not match the current index then swap those values
 6. Increase the starting index by 1
 7. Repeat 2 through 6 until the starting index >= length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

Sorting Algorithms

- **Problem:**
– Given any array of integers, develop an algorithm that sorts the values from smallest to largest.
- **Selection Sort**
1. Start from index 0
 2. Assume the starting index has the smallest value and record that index
 3. Sequentially check every other value
 4. If a value is found that is smaller at another index, then record that current index
 5. Once all values have been checked if the recorded index does not match the current index then swap those values
 6. Increase the starting index by 1
 7. Repeat 2 through 6 until the starting index >= length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

Sorting Algorithms

- **Problem:**
– Given any array of integers, develop an algorithm that sorts the values from smallest to largest.
- **Selection Sort**
1. Start from index 0
 2. Assume the starting index has the smallest value and record that index
 3. Sequentially check every other value
 4. If a value is found that is smaller at another index, then record that current index
 5. Once all values have been checked if the recorded index does not match the current index then swap those values
 6. Increase the starting index by 1
 7. Repeat 2 through 6 until the starting index >= length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

Sorting Algorithms

- **Problem:**
– Given any array of integers, develop an algorithm that sorts the values from smallest to largest.
- **Selection Sort**
1. Start from index 0
 2. Assume the starting index has the smallest value and record that index
 3. Sequentially check every other value
 4. If a value is found that is smaller at another index, then record that current index
 5. Once all values have been checked if the recorded index does not match the current index then swap those values
 6. Increase the starting index by 1
 7. Repeat 2 through 6 until the starting index >= length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

Sorting Algorithms

•Problem:
—Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

•Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

↑

61

Sorting Algorithms

•Problem:
—Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

•Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

↑

62

Sorting Algorithms

•Problem:
—Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

•Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

↑

63

Sorting Algorithms

•Problem:
—Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

•Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

↑

64

Sorting Algorithms

•Problem:
—Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

•Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

↑

65

Sorting Algorithms

•Problem:
—Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

•Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

↑

66

Sorting Algorithms

Problem:
 -Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

67

Sorting Algorithms

Problem:
 -Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	10	8	7	6	12	5	11	9

68

Sorting Algorithms

Problem:
 -Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	5	8	7	6	12	10	11	9

69

Sorting Algorithms

Problem:
 -Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	5	8	7	6	12	10	11	9

70

Sorting Algorithms

Problem:
 -Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	5	8	7	6	12	10	11	9

71

Sorting Algorithms

Problem:
 -Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	5	8	7	6	12	10	11	9

72

Sorting Algorithms

Problem:
—Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	5	8	7	6	12	10	11	9

73

Sorting Algorithms

Problem:
—Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	5	6	7	8	12	10	11	9

74

Sorting Algorithms

Problem:
—Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

Example

Index	0	1	2	3	4	5	6	7
Value	5	6	7	8	12	10	11	9

75

A Few Swaps Later

76

Sorting Algorithms

Problem:
—Given any array of integers, develop an algorithm that sorts the values from smallest to largest.

Selection Sort

1. Start from index 0
2. Assume the starting index has the smallest value and record that index
3. Sequentially check every other value
4. If a value is found that is smaller at another index, then record that current index
5. Once all values have been checked if the recorded index does not match the current index then swap those values
6. Increase the starting index by 1
7. Repeat 2 through 6 until the starting index \geq length

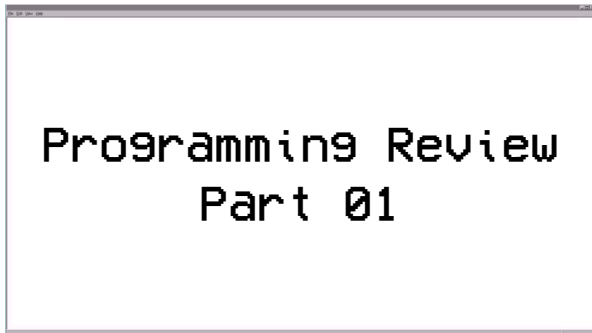
Example

Index	0	1	2	3	4	5	6	7
Value	5	6	7	8	9	10	11	12

77

Selection Sort Implementation

78



79