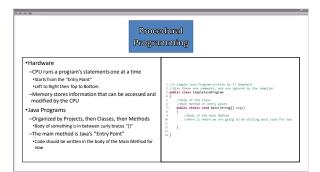


Basic Computation Part 01

2

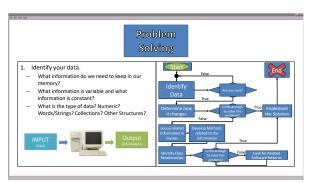
1

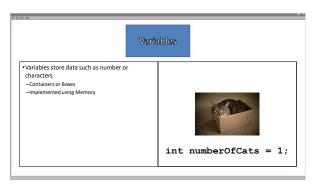


Problem
Solving

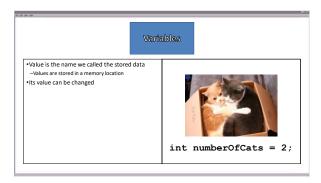
1. Identify your Data.
2. Determine how the data changes over time.
3. Consider structures for both behavior and data.
4. Group together (encapsulate) related information into Classes of Objects.
5. Develop functionality / methodologies that relates to the behavior of your Objects.
6. Further identify relationships between the Classes and optimize the structure.
7. Determine if there exists software patterns that may assist.

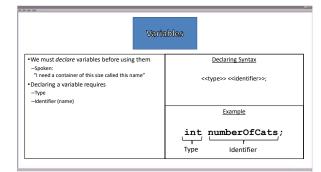
3 4





5 6

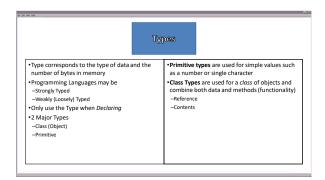




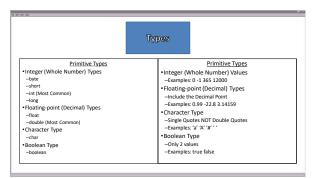
8

7

Types



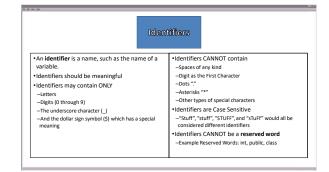
9 10





11 12





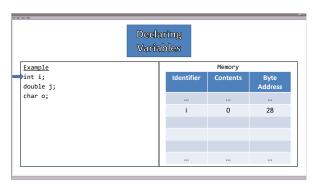
13 14

lden	iifiers
Naming Conventions	Good Examples
Class Types start with an Uppercase character	int test01;
-Example: String	double largeValues;
• Primitive Types start with a Lowercase character —Example: int	boolean inClass;
•Variables identifiers of both start with a Lowercase Character	Bad Examples
•Multiword identifiers are "punctuated" using	int 1Test;//Started with a digit
uppercase characters	double big vals;//Used a space boolean class;//Class is a reserved word

	Decla Varia	aring ıbles		
Example			Memory	
int i; double j;		Identifier	Contents	Byte Address
char o;				
				28

15 16

	Decla Varia	ring bles		
Example			Memory	
int i; double j;		Identifier	Contents	Byte Address
char o;				
				28



17 18



Declaring Variables

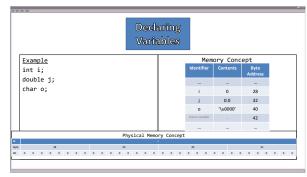
Example int i; | Identifier | Contents | Byte Address | Contents |

19 20

	Decla Varia	aring Ibles		
Example			Memory	
int i; double j;		Identifier	Contents	Byte Address
char o;				
		i	0	28
		j	0.0	32

21 22

	Decla Varia	aring ables		
Example			Memory	
<pre>int i; double j;</pre>		Identifier	Contents	Byte Address
char o;				
		i	0	28
		j	0.0	32
		0	'\u0000'	40
		???	???	42



23 24

Assi9nin9 Values

Assignment
Operator

- The equals symbol "=" is the assignment operator
- Stores values found on the right hand side (RHS) of the operator into the identifier found on the left hand side (IHS)
- Assignments are valid if the type matches are is at least compatible
- Primitive types can be stored in other primitive types as long the type's byte amount it less than or equal to value being stored
- Otherwise "type casting" is required
- Type casting does not round it cuts off everything past the decimal point ""
- Spoken:
- "Store this value in this container"

Assignment

- (<identifier>> = <<value>>;

- (xidentifier>> = <<value>>;

- (xidentifi

25 26

Assign Open	rator		
Declare and assigning initial values		Memory	
-Good programming practice to assign initial values -Shortens two statements into one -Types are not still used after the declaration	Identifier	Contents	Byte Address
-,,			
Example int i = 0;			
double j = 22.3;			
char o = 'h'; i = (int)j;			

Assignment
Operator

Declare and assigning initial values
Good programming practice to assign initial values
Shorters two statements into one
Types are not still used after the declaration

Example
Int i = 0;
double j = 22.3;
char o = 'h';
i = (int)j;

Memory
Identifier Contents Byte
Address
...
...
...

27 28

Assign Oper	ment ator		
Declare and assigning initial values		Memory	
-Good programming practice to assign initial values -Shortens two statements into one	Identifier	Contents	Byte Address
-Types are not still used after the declaration			
	i	0	28
Example int i = 0;			
double j = 22.3;			
char o = 'h';			
i = (int)j;			

Assignment
Operator

• Declare and assigning initial values
—Good programming practice to assign initial values
—Shortens two statements into one
—Types are not still used after the declaration

Example
—Init i = 0;
double j = 22.3;
char o = 'h';
i = (int)j;

...

29 30

Assign Oper	ment ator		
Declare and assigning initial values		Memory	
-Good programming practice to assign initial values -Shortens two statements into one -Types are not still used after the declaration	Identifier	Contents	Byte Address
- Types are not still used after the declaration			
	i	0	28
<u>Example</u>			
int i = 0;			
double j = 22.3;			
char o = 'h';			
i = (int)j;			

Assignment
Operator

-Declare and assigning initial values
-Good programming practice to assign initial values
-Shortens two statements into one
-Types are not still used after the declaration

Example int i = 0;
| double j = 22.3; | char o = 'h'; | i = (int)j; |

31 32

Assig Ope	rator			
Declare and assigning initial values			Memory	
-Good programming practice to assign initial values -Shortens two statements into one -Types are not still used after the declaration	Ide	ntifier	Contents	Byte Address
- Types are not still used after the declaration				
		i	0	28
<u>Example</u>		j	22.3	32
int i = 0; double j = 22.3;				
char o = 'h';				

33 34

Assign Open	nment rator		
Declare and assigning initial values		Memory	
-Good programming practice to assign initial values -Shortens two statements into one	Identifier	Contents	Byte Address
-Types are not still used after the declaration			
	i	0	28
Example	j	22.3	32
int i = 0; double j = 22.3;	0	'h'	40
char <mark>o = 'h'</mark> ; i = (int)j;			

35 36

Assign Oper	nment		
Declare and assigning initial values		Memory	
-Good programming practice to assign initial values -Shortens two statements into one	Identifier	Contents	Byte Address
-Types are not still used after the declaration			
	i	0	28
<u>Example</u>	i	22.3	32
int i = 0; double j = 22.3;	0	'h'	40
char o = 'h';			
i = (int)j;			

Operator Declare and assigning initial values Memory -Good programming practice to assign initial values -Shortens two statements into one Contents -Types are not still used after the declaration i 22 28 Example j 22.3 32 int i = 0; 'h' 40 0 double j = 22.3;
char o = 'h'; i = (int)j;

38

37

Assign	nment rator		
Declare and assigning initial values		Memory	
Good programming practice to assign initial valuesShortens two statements into oneTypes are not still used after the declaration	Identifier	Contents	Byte Address
	i	22	28
<u>Example</u>	j	22.3	32
int i = 0; double j = 22.3;	o	'h'	40
char o = 'h';			
i = (int)j;			

-Establishes a value that cannot change
-MUST assign a value initially
-Great for avoiding "magic numbers"
-Good programming practice
-Make the scope public
-Make it static
-Capitalize all characters in the identifier

-Examples
-public static final int BOARD_SIZE = 10;

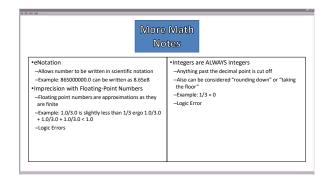
39 40

	lath erators
•Performs computation and then assigns the results	Syntax
•Order of Operations	< <identifier>> = <<value>> <<operator>> <<value>>;</value></operator></value></identifier>
•Basic Math Operations -Addition "+" -Subtraction "-"	
-Multiplication "*"	Examples
-Division "/"	//Variables
• Mod Operator "%" —Returns the remainder after division —Ex: 15 % 2 = 1	<pre>int value = 64 % i + 32; //Constants public static final double PI = 3.14159; public static final double PI_SQ = PI*PI;</pre>

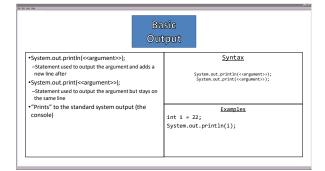
*Compute and Assign (C&A) Operators

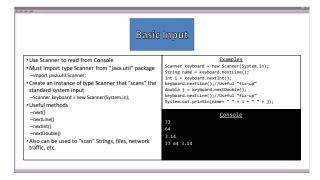
-Shorthand for applying some operator and value to a variable
-Same as:
-Common Versions

41 42



43 44





45 46