

# CMPSC 100

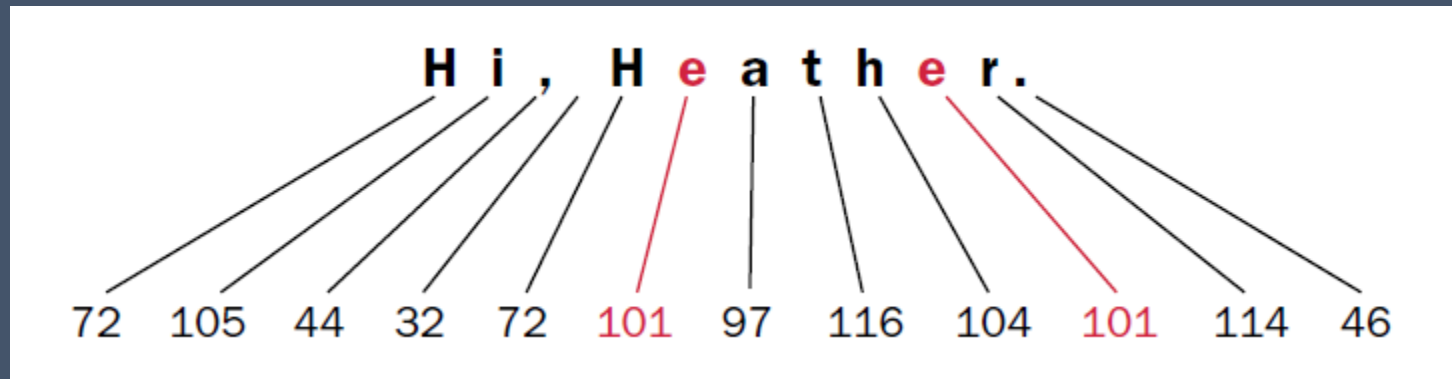
Computational Expression

# Primitive data types

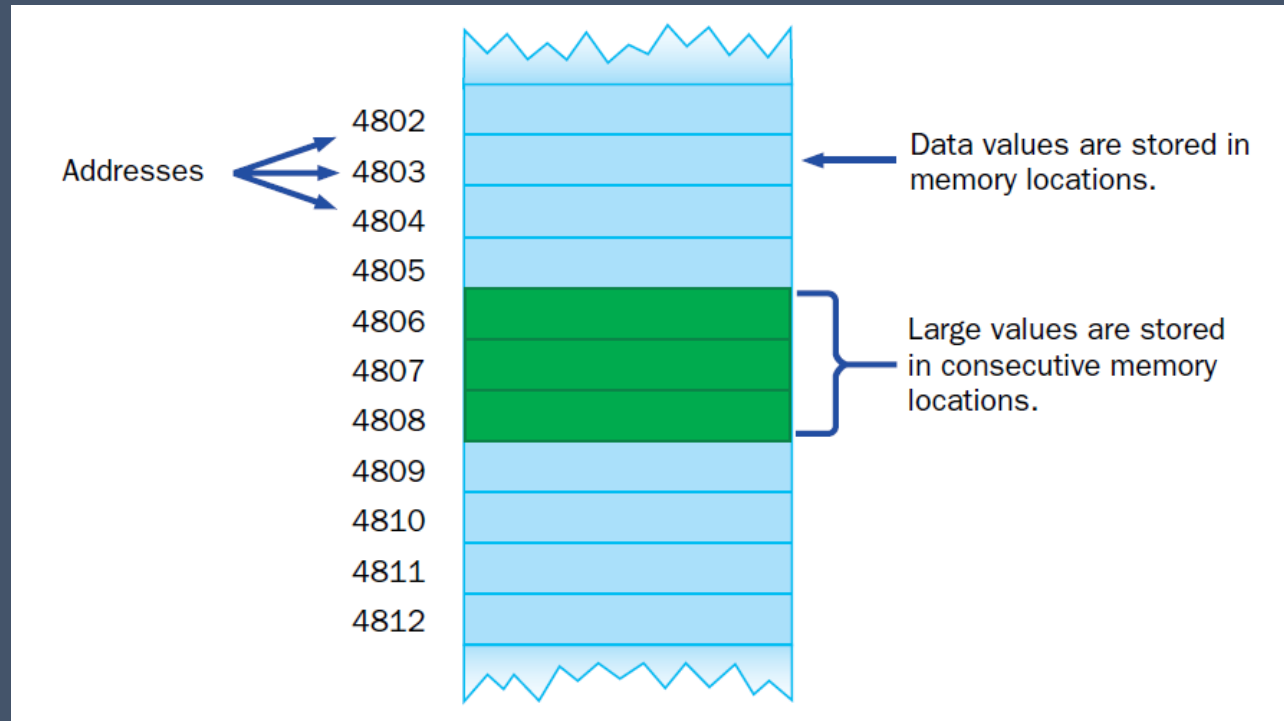
Type	Storage	Min Value	Max Value
byte	8 bits	-128	127
short	16 bits	-32,768	32,767
int	32 bits	-2,147,483,648	2,147,483,647
long	64 bits	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
float	32 bits	Approximately $-3.4\text{E}+38$ with 7 significant digits	Approximately $3.4\text{E}+38$ with 7 significant digits
double	64 bits	Approximately $-1.7\text{E}+308$ with 15 significant digits	Approximately $1.7\text{E}+308$ with 15 significant digits

# Primitive data type

- “String” is missing, though
  - Why?
  - Because Strings are essentially made up of “Char” objects (which aren’t primitives either)
  - And because “Char” objects are pointers to numbers representing elements of a set



# Java as a “strongly-typed” language



# R to L, L to Right “association”

Precedence Level	Operator	Operation	Associates
1	+	unary plus	R to L
	—	unary minus	
2	*	multiplication	L to R
	/	division	
	%	remainder	
3	+	addition	L to R
	—	subtraction	
	+	string concatenation	
4	=	assignment	R to L

# int vs. float vs. double

$$\frac{22}{7} = 3.\overline{142857},$$
$$\pi = 3.14159265\dots$$

```
int pi = 22/7  
short pi = 22/7  
long pi = 22L/7L
```

$\pi$  3  
 $\pi$  3  
 $\pi$  3

```
float pi = 22f/7f
```

3.14285  
7  
 $\pi$

```
double pi = 22/7
```

3.14285714285714  
3  
 $\pi\pi$

# Let's try a word problem!

Ten thieves stole some jewels from a local shop. They stole a bag of fifty six pearls. Some of them were armed and some were unarmed. The armed ones were those of senior rank. When it came to dividing them up, each senior robber took six pearls, and each junior robber got five. How many of the robbers were senior?





```
package wordproblems;

/** This class solves the Thieves' Story.
 *
 * @author Alton Caylor
 */
public class ThievesStory { final static int jewels = 56;
    final static int thieves = 10;
    final static int juniorShare = 5;
    final static int seniorShare = 6; /** This is the entry point.
 *
 * @param The command line arguments.
 */
    public static void main(String[] args) {
        int seniorMembers = (juniorShare*thieves-jewels)/(juniorShare-seniorShare);
        int juniorMembers = (seniorShare*thieves-jewels)/(seniorShare-juniorShare);
        System.out.print("There are " + juniorMembers + " junior members, " +
            "and " + seniorMembers + " senior members.");
    }
}
```