Data Types

- declare variables – variables must be declared before they are used

public static int x; public → accessible by programs outside of the class.

private static int y, z; private → hides from classes within the same package (group of classes)

protected static int y, z; protected → form of private accessible only by the class and sub classes

static → all instances of the class (all objects created from the class) will have the same value for this parameter/characteristic/data... more on this later

```
boolean gameOver;
char answer;
int noOfStudents;
double totalPrice, salesTax;

int total = 0;
```

- primitive data types

Type	**Description**	**Default**	**Size**	**Example**	**Range of Numeric Data Types (min value, max value)**
boolean	true or false	False	1 bit		Only two values possible; 0 or 1
byte	Signed integer	0	8 bits		1000 000 = -128 (-2^7) 0111 111 = 127 ($2^7 - 1$)
char	Unicode character	0	16 bits		
short	Signed integer	0	16 bits		
int	Signed integer	0	32 bits		
long	Signed integer	0	64 bits		
float	IEEE 754 floating point	0.0	32 bits		See IEEE 754
double	IEEE 754 floating point	0.0	64 bits		See IEEE 754

- “Final” data types –

Arithmetic – when assigning values, variable should be the same data type as data being assigned.

Example:

```
private static int x;

x = 6.5/2;    (error since 6.5/2 is a float data type)
```

+ addition
– subtraction
* multiplication
/ division
% modulus

Expression	Result	Type of Result
`(3+7)*(4-9)`	-50	int
`3.0+7*4-9`	22	double
`56 / 15`	3	int
`56 % 15`	11	int
`2 / 3 + 4 / 5`	0	int
`(2 * 15 + 7) % 5`	2	int
`2.3 * 7 - 20.5`	-5.3	double

++ increment and -- decrement

```
int myVariable = 10;     x = myVariable++;     // x will equal 10

int myVariable = 10;     x = ++myVariable;     // x will equal 11
```

Converting data types

“Widening” – data is converted and no data is lost for example byte to integer; this is done automatically

Byte (represented with 8 bits) → convert to → integer (represented with 32 bits); widening because going from 8 bits to 32 bits.

Example:

```
int a = 100;
long  b = a;
System.out.println(b);
```

“Narrowing” – data conversion may lose data for example converting integer to short; this is allowed only by “casting” data to another type.

Narrowing because you are converting from integer (32 bits) to short (16 bits).

Example:

```
int  a = 1061;      // try 2093, 293, 549, 1061
short  b = short(a);
System.out.println(a);
System.out.println(b);
```

go to website   http://mrbool.com/java-data-type-conversion/29257

and read and understand 'Widening Conversion' and 'Narrowing Conversion',   stop at 'Automatic Conversion'

- Object or reference data types
    - String data type
    - Memory location

      ```
      String name;
      new String("Jack Bauer");
      ```

- 3.4 formatting – skip this section
- Arithmetic expressions
    - Integer ÷ integer = ????
    - Assigning values to correct variable types
        - myVariable = 12 / 7
- Data type Conversions
    - Promotion
    - Casting – what happens to information?

Page 32 Questions page 32 questions 1, 3, 4

Name:

What does the “final” do in the statement:

```
final   int   x = 12;
```

What is the difference between “int” and “long”?

Use one short sentence to describe “promotion” when converting data types.

Name:

What does the “final” do in the statement:

final   int   x = 12;

What is the difference between “int” and “long”?

Use one short sentence to describe “promotion” when converting data types.