

Unit Two: Fundamental Data Types & Text Files

Topic 1: What is an identifier?

An identifier is the name of a variable, constant, or method.

- These identifiers can vary from one to several characters. The first character must be a letter or an underscore, and subsequent characters must be either letters, digits, or underscores.
- Identifiers may be of any length, but not all the characters will necessarily be significant. At least 6 characters will always be significant.

examples:

```
correct:    count  test3  new_total
incorrect:  2count  old!sum  new...total
```

Topic 2: What is a constant?

A constant is a named location in memory that is used to hold a value that can not be modified by the program. All constants must be declared before they can be used.

When used locally, the declaration would look like this...

```
final double PI = 3.14;
```

The reserved word `final`, indicates the double value held in a memory location called `pi` can not change.

Frequently, a constant value is needed in several methods. Then you need to declare it together with the instance variables of the class. The declaration would look like this...

```
private static final double PI = 3.14;
```

As a matter of style, constants use all upper case letters, so they are quickly visually distinguishable from variables.

Constants should be used liberally as they aid in the readability and modifiability of programs.

Topic 3: What is a variable?

A variable is a named location in memory that is used to hold a value that may be modified by the program. All variables must be declared before they can be used (not necessarily all at once at the beginning of the program). The general form of a declaration is...

```
type variable list;
```

```
examples:  int s1, s2, s3;
           double ave;
           boolean flag;
```

Topic 4: What are the Arithmetic Operators of Java?

Operator	Action
-	subtraction
+	addition
*	multiplication
/	division
%	modulus (remainder in long division)
--	decrement
++	increment

Topic 5: What are the results of general operations?

	double	int	int
	double	double	int
-	double	error*	int
+	double	error*	int
*	double	error*	int
/	double	error*	int ** Not what you'd think!
%	double	double	int

* These operations can work if the result is placed in a memory cell of type double. However, if the result is placed in a memory cell of type int an error message will result.

** WARNING: In Java an integer divided by an integer is an integer.

```
examples:  int x = 12/5;           x contains 2
           double x = 12/5;       x contains 2.0
```

Topic 6: What does it mean to Cast an expression?

It forces an expression to be of a specific type.

sample program:

```
public static void main(String [] args)
{
    int a = 5;
    int b = 2;
    double x;

    x = a/b;                // No Casting
    System.out.println(x);

    x = (double) a/b;      // Casting the expression as a
    System.out.println(x); // double
}
```

Output:

```
2
2.5
```

Topic 7: Formatting Numbers - not tested on AP Exam

Java 1.5.0 includes the method *printf* for formatting output. The parameters of the *printf* method consist of a format string and one or more data values, according to the following general form:

```
printf(<format string>, <data value 1>, ..., <data value n>)
```

The format string is a combination of literal string information and formatting information. The formatting information consists of one or more format specifiers. These codes begin with a '%' character and end with a letter that indicates the format type. Inside each specifier there will be multiple format flags.

Format Types

code	type	example
----	----	-----
d	Base 10 integer	123
f	Fixed floating point	12.30
s	String	Tax:
n	end line	

Format Flags

flag -----	meaning -----	example -----
-	left justification	1.23 followed by spaces
0	show leading zeros	0930
+	show + sign	+1.23
,	show decimal separators	12,300
\$	show dollar sign	\$56.70
%	show percent sign	5.8%

Examples:

- 1)

```
double total = 3.5;
System.out.printf("Total:%5.2f%n%n", total);
```
- 2)

```
double tax = 0.2875;
System.out.printf("%-6s%5.2f%n%n", "Tax", tax);
```
- 3)

```
double dollars = 25;
double tax = dollars * 0.125;
System.out.printf("Income:  $%.2f%n", dollars);
System.out.printf("Tax owed:  $%.2f%n%n", tax);
```
- 4)

```
int x = 34;
double y = 56.7;
System.out.printf("%4d%7.2f%n", x, y);
System.out.printf("%-4d$%8.3f%n", x, y);
```

Output:

- ```
1234567890123456789
```
- 1) Total: 3.50
  - 2) Tax: 0.29
  - 3) Income: \$25.00  
Tax owed: \$3.13
  - 4) 34 56.70  
34 \$ 56.700

## Topic 8: Parametric Constructors

Let's go back to the Cylinder class example from last unit.

```
public class Cylinder
{
 private int height; // instance fields
 private int radius;

 public Cylinder() // default constructor - used to
 { // initialize instance fields with
 height = 5; // values decided on by the
 radius = 2; // programmer of the class
 }

 public Cylinder(int h, int r) // parametric constructor - used
 { // to initialize instance fields
 height = h; // with values decided on by the
 radius = r; // user of the class
 }
 .
 .
 .
}

public class Driver
{
 public static void main(String[] args)
 {
 Cylinder x = new Cylinder(8,3); // user of the Cylinder
 . // class is calling the
 . // parametric constructor
 . // by constructing an
 // object of type
 // Cylinder with two
 // arguments 8 & 3
 }
}
```

Assignment U2A1: The Product Class Project