Introduction to Computer Programming

Lecture 2- Primitive Data and Stepwise Refinement

Data Types

• **Type** - A category or set of data values.
  – Constrains the operations that can be performed on data
  – Many languages ask the programmer to specify types
  – Examples: integer, real number, string
How Does the Computer Store Data?

• Internally, computers store everything as 1s and 0s
  
  $104 \rightarrow 01101000$
  
  "hi" $\rightarrow 01101000110101$

Java's Primitive Types

• **Primitive Types** - 8 simple types for numbers, text, etc.
• Java also has **object types**, which we'll talk about later
Data Types in Java

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>Integers (up to 2^31-1)</td>
<td>42, -3, 0, 926394</td>
</tr>
<tr>
<td>double</td>
<td>Real numbers (up to 10^{308})</td>
<td>3.1, -0.25, 9.4e3</td>
</tr>
</tbody>
</table>
| char  | Single text characters   | 'a', 'X', '?', '
'   |
| boolean| Logical values         | true, false          |

Expressions

- **Expression** - A value or operation that computes a value.

- Examples:  
  \[ 1 + 4 \times 5 \]
  \[ (7 + 2) \times 6 / 3 \]
  \[ 42 \]

- The simplest expression is a *literal value*.
- A complex expression can use operators and parentheses.
Arithmetic Operators

• **Operator**: Combines multiple values or expressions.
  
  +  addition
  -  subtraction (or negation)
  *  multiplication
  /  division
  %  modulus (a.k.a. remainder)

Evaluating Expressions

• As a program runs, its expressions are *evaluated*.

• 1 + 1 evaluates to 2

• `System.out.println(3 * 4);` prints 12

• How would we print the text 3 * 4?
• When we divide integers, the quotient is also an integer.
  – \(14/4\) is 3, not 3.5

<table>
<thead>
<tr>
<th>4</th>
<th>3 (\underline{14})</th>
<th>10 (\underline{45})</th>
<th>27 (\underline{1425})</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>12</td>
<td>40</td>
<td>135</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>5</td>
<td>75</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>54</td>
<td>54</td>
</tr>
</tbody>
</table>

• More examples:
  \(32/5\) is 6
  \(84/10\) is 8
  \(156/100\) is 1

• Dividing by 0 causes an error when your program runs.
Integer Remainder With %

• The % operator computes the remainder from integer division.

\[
\begin{align*}
14 \% 4 & \text{ is } 2 \\
218 \% 5 & \text{ is } 3
\end{align*}
\]

What is the result?

\[
\begin{align*}
45 \% 6 & \\
2 \% 2 & \\
8 \% 20 & \\
11 \% 0 &
\end{align*}
\]

Applications Of % Operator

• Obtain last digit of a number:

\[
230857 \% 10 \text{ is } 7
\]

• Obtain last 4 digits:

\[
658236489 \% 10000 \text{ is } 6489
\]

• See whether a number is odd:

\[
\begin{align*}
7 \% 2 & \text{ is } 1 \\
42 \% 2 & \text{ is } 0
\end{align*}
\]
Precedence

• **Precedence** - Order in which operators are evaluated.

• Generally operators evaluate left-to-right.
  \[ 1 - 2 - 3 = (1 - 2) - 3 = -4 \]

• But \( * \) \( / \) have a higher level of precedence than \( + \)
  \[-1 + 3 \times 4 = 13 \]

\[
\begin{align*}
6 + 8 / 2 \times 3 &= 13 \\
6 + 4 \times 3 &= 18
\end{align*}
\]

Precedence and Parentheses

• Parentheses can force a certain order of evaluation:
  \[ (1 + 3) \times 4 = 16 \]

• Spacing does not affect order of evaluation
  \[ 1 + 3 \times 4 - 2 = 11 \]
Precedence Examples

1 \* 2 + 3 \* 5 \% 4

1 + 8 \% 3 \* 2 - 9

Precedence Questions

• What values result from the following expressions?
  
  9 / 5
  695 \% 20
  7 + 6 \* 5
  7 \* 6 + 5
  248 \% 100 / 5
  6 \* 3 - 9 / 4
  (5 - 7) \* 4
  6 + (18 \% (17 - 12))
Real Numbers (Type `double`)

- Examples:
  6.022, -42.0, 2.143e17
- Placing .0 or . after an integer makes it a `double`.
  62.0  13.  .5
- The operators `+ - * / % ()` all still work with `double`.
  `/` produces `15.0 / 2.0 = 7.5`
- Precedence is the same: `()` before `* / %` before `+ -`

Real Number Example

```
2.0 * 2.4 + 2.25 * 4.0 / 2.0

4.8 + 2.25 * 4.0 / 2.0

4.8 + 9.0 / 2.0

4.8 + 4.5

9.3
```
Mixing Types

• When `int` and `double` are mixed, the result is a `double`.
  
  \[
  4.2 \times 3 \text{ is } 12.6
  \]

• The conversion is per-operator, affecting only its operands.

\[
\begin{align*}
7 / 3 \times 1.2 + 3 / 2 &= 2.0 + 10 / 3 \times 2.5 - 6 / 4 \\
2 \times 1.2 + 3 / 2 &= 2.0 + 7.5 - 6 / 4 \\
2.4 + 3 / 2 &= 2.0 + 7.5 - 6 / 4 \\
2.4 + 1 &= 2.0 + 7.5 - 1 \\
3.4 &= 2.0 + 7.5 - 1
\end{align*}
\]

3 / 2 is 1 above, not 1.5
String Concatenation

- **String concatenation** - Using + between a string and another value to make a longer string.

  - "hello" + 42 is "hello42"
  - 1 + "abc" + 2 is "1abc2"
  - "abc" + 1 + 2 is "abc12"
  - 1 + 2 + "abc" is "3abc"
  - "abc" + 9 * 3 is "abc27"
  - "1" + 1 is "11"
  - 4 - 1 + "abc" is "3abc"

- Use + to print a string and an expression's receipt.

Receipt Example

What's wrong with this program?

```java
public class Receipt {
    public static void main(String[] args) {
        // Calculate total owed, assuming 8% tax / 15% tip */
        System.out.println("Subtotal:");
        System.out.println(38 + 40 + 30);
        System.out.println("Tax:");
        System.out.println((38 + 40 + 30) * .08);
        System.out.println("Tip:");
        System.out.println((38 + 40 + 30) * .15);
        System.out.println("Total:");
        System.out.println(38 + 40 + 30 +
                (38 + 40 + 30) * .08 +
                (38 + 40 + 30) * .15);
    }
}
```

(38 + 40 + 30) is repeated

So many println statements
Variables

- **variable**: A piece of the computer's memory that is given a name and type, and can store a value.
  - Like preset stations on a car stereo, or cell phone speed dial:

![Car stereo and cell phone]

Declaration

- **Variable declaration** - Sets aside memory for storing a value.
- Variables must be declared before they can be used.
- Syntax:

  ```
  DataType VariableName;
  ```
  - The variable's name is an *identifier*.

  ```
  int x;
  ```

  ```
  double myGPA;
  ```
Assignment

- **Assignment** - Stores a value into a variable.
- The value can be an expression; the variable stores its result.
- Syntax:
  
  ```java
  VariableName = expression;
  ```

  ```java
  int x;
  x = 3;
  double myGPA;
  myGPA = 1.0 + 2.25;
  ```

Using Variables

- Once given a value, a variable can be used in expressions:

  ```java
  int x;
  x = 3;
  System.out.println("x is " + x); // x is 3
  System.out.println(5 * x - 1); // 5 * 3 - 1
  ```

- You can assign a value more than once:

  ```java
  int x;
  x = 3;
  System.out.println(x + " here"); // 3 here
  x = 4 + 7;
  System.out.println("now x is " + x); // now x is 11
  ```
Declaration/Initialization

- A variable can be declared/initialized in one statement.
- Syntax:
  \[ \text{DataType VariableName} = \text{value}; \]
  
  \[
  \begin{array}{c|c}
  \text{myGPA} & 3.95 \\
  \end{array}
  \]

  \[
  \begin{array}{c|c}
  \text{x} & 14 \\
  \end{array}
  \]

Assignment and Algebra

- Assignment uses =, but it is not an algebraic equation.
- = means, "store the value at right in variable at left"
- The right side expression is evaluated first, and then its result is stored in the variable at left.
- What happens here?
  \[
  \begin{array}{c|c}
  \text{x} & 3 \\
  \end{array}
  \]

  \[
  \begin{array}{c|c}
  \text{x} & 5 \\
  \end{array}
  \]
Assignment and Types

• An int value can be stored in a double variable.
• The value is converted into the equivalent real number.
  \[
  \text{double myGPA = 4;}
  \]
  
  \[
  \text{double avg = 11 / 2;}
  \]

• Why does avg store 5.0 and not 5.5?

Compiler Errors

• A variable can't be used until it is assigned a value.
  ```java
  int x;
  System.out.println(x);
  // ERROR: x has no value
  ```

• You may not declare the same variable twice.
  ```java
  int x;
  int x; // ERROR: x already exists
  ```

  ```java
  int x = 3;
  int x = 5; // ERROR: x already exists
  ```

• How can this code be fixed?
### Printing A Variable's Value

- Use `+` to print a string and a variable's value on one line.

  ```java
  double grade = (95.1 + 71.9 + 82.6) / 3.0;
  System.out.println("Your grade was "+ grade);
  ```

  ```java
  int students = 11 + 17 + 4 + 19 + 14;
  System.out.println("There are "+ students + " students in the course.");
  ```

- Output:

  Your grade was 83.2
  There are 65 students in the course.

### Receipt Question

Improve the receipt program using variables.

```java
public class Receipt {
  public static void main(String[] args) {
    // Calculate total owed,
    // assuming 8% tax / 15% tip
    System.out.println("Subtotal:");
    System.out.println(38 + 40 + 30);
    System.out.println("Tax:");
    System.out.println((38 + 40 + 30) * .08);
    System.out.println("Tip:");
    System.out.println((38 + 40 + 30) * .15);
  }
}
```
System.out.println("Total:");
System.out.println(38 + 40 + 30 +
    (38 + 40 + 30) * .15 +
    (38 + 40 + 30) * .08);
}
}
Finding the Average

- Let’s write a program that will find the average any 3 numbers we try:
- We now need to:
  1. Find our three values
  2. Add the values
  3. Divide the sum by 3
  4. Print the result

Writing Average3b

This first step becomes:

1.1 Find the first value
1.2 Find the second value
1.3 Find the third value
2. Add the values
3. Divide the sum by 3
4. Print the result
Writing Avg3 (continued)

Since we want the computer to print out some kind of prompt, the first step becomes:

1.1.1 Prompt the user for the first value
1.1.2 Read in the first value
1.2.1 Prompt the user for the second value
1.2.2 Read in the second value
1.3.1 Prompt the user for the third value
1.3.2 Read in the third value
2. Add the values
3. Divide the sum by 3
4. Print the result

Writing Avg3 (continued)

We can prompt the user with:

1.1.1 System.out.println
       ("Enter the first value ?");
1.1.2 Read in the first value
1.2.1 System.out.println
       ("Enter the second value ?");
1.2.2 Read in the second value
1.3.1 System.out.println
       ("Enter the third value ?");
1.3.2 Read in the third value
2. Add the values
3. Divide the sum by 3
4. Print the result
The **Scanner** Class

- Most programs will need some form of input.
- At the beginning, all of our input will come from the keyboard.
- To read in a value, we need to use an object belonging to a class called Scanner:
  
  ```java
  Scanner keyb = new Scanner(System.in);
  ```

---

**Reading from the keyboard**

- Once we declare keyb as Scanner, we can read integer values by writing:
  
  ```java
  variable = keyb.nextInt();
  ```
Writing the input statements in Average3b

We can read in a value by writing:

```java
System.out.println
("What is the first value\t?";
int value1 = keyb.nextInt();
System.out.println
("What is the second value\t?");
int value2 = keyb.nextInt();
System.out.println
("What is the third value\t?");
int value3 = keyb.nextInt();
```

2. Add the values
3. Divide the sum by 3
4. Print the result

Writing the assignments statements in Average3b

```java
System.out.println
("What is the first value\t?");
int value1 = keyb.nextInt();
System.out.println
("What is the second value\t?");
int value2 = keyb.nextInt();
System.out.println
("What is the third value\t?");
int value3 = keyb.nextInt();
```

```
sum = value1 + value2
\_ + value3;
```

3. Divide the sum by 3
4. Print the result

*Adding up the three values*
Writing the assignments statements in Average3b

```
System.out.println
    ("What is the first value\t?");
int value1 = keyb.nextInt();
System.out.println
    ("What is the second value\t?");
int value2 = keyb.nextInt();
System.out.println
    ("What is the third value\t?");
int value3 = keyb.nextInt();
sum = value1 + value2 + value3;

average = sum / 3;
```

4. Print the result

```
Calculating the average
```

Writing the output statement in Average3b

```
System.out.println
    ("What is the first value\t?");
int value1 = keyb.nextInt();
System.out.println
    ("What is the second value\t?");
int value2 = keyb.nextInt();
System.out.println
    ("What is the third value\t?");
int value3 = keyb.nextInt();
sum = value1 + value2 + value3;
average = sum / 3;
System.out.println("The average is "+ average);
```
import java.util.Scanner;

public class Average3b {
    public static void main(String[] args) {
        int sum, average;
        Scanner keyb = new Scanner(System.in);

        System.out.println("What is the first value\t?\t");
        int value1 = keyb.nextInt();

        System.out.println("What is the second value\t?\t");
        int value2 = keyb.nextInt();

        System.out.println("What is the third value\t?\t");
        int value3 = keyb.nextInt();

        sum = value1 + value2 + value3;
        average = sum / 3;
        System.out.println("The average is \t" + average);
    }
}

Another example – calculating a payroll

- We are going to write a program which calculates the gross pay for someone earning an hourly wage.
- We need two pieces of information:
  - the hourly rate of pay
  - the number of hours worked.
- We are expected to produce one output: the gross pay, which we can find by calculating:
  - Gross pay = Rate of pay * Hours Worked

Our Design for payroll

1. Get the inputs
2. Calculate the gross pay
3. Print the gross pay

*We can substitute:*

  1.1 Get the rate
  1.2 Get the hours
Developing The Payroll Program

1. Get the rate
   - 1.1 Prompt the user for the rate
   - 1.2 Read the rate
2. Get the hours
   - 1.2.1 Prompt the user for the hours
   - 1.2.2 Read the hours
3. Calculate the gross pay
4. Print the gross pay

We can substitute

Coding the payroll program

- Before we code the payroll program, we recognize that the values (rate, hours and gross) may not necessarily be integers.
- We will declare these to be double, which means that they can have (but do not have to have) fractional parts.
- In Java, we usually declare our variables where they first appear in the program.
Developing The Payroll Program (continued)

1.1.1 Prompt the user for the rate
1.1.2 Read the rate
1.2.1 Prompt the user for the hours
1.2.2 Read the hours
2. Calculate the gross pay
3. Print the gross pay

```java
System.out.println("What is your hourly pay rate?");
double rate = keyb.nextDouble();
```

Developing The Payroll Program (continued)

```java
System.out.println("What is your hourly pay rate?");
double rate = keyb.nextDouble();
1.2.1 Prompt the user for the hours
1.2.2 Read the hours
2. Calculate the gross pay
3. Print the gross pay

System.out.println("How many hours did you work?");
double hours = keyb.nextDouble();
```
Developing The Payroll Program (continued)

```java
System.out.println("What is your hourly pay rate?");
double rate = keyb.nextDouble();
System.out.println("How many hours did you work?");
double hours = keyb.nextDouble();
```

2. Calculate the gross pay
3. Print the gross pay

```
double gross = rate * hours;
```

Developing The Payroll Program (continued)

```java
System.out.println("What is your hourly pay rate?");
double rate = keyb.nextDouble();
System.out.println("How many hours did you work?");
double hours = keyb.nextDouble();

double gross = rate * hours;
```

3. Print the gross pay

```
System.out.println("Your gross pay is ");
System.out.println(gross);
```
import java.util.Scanner;

public class Payroll {

    // This program calculates the gross pay for an hourly worker
    // Inputs - hourly rate and hours worked
    // Output - Gross pay
    public static void main(String[] args) {
        Scanner keyb = new Scanner(System.in);

        // Get the hourly rate
        System.out.println("What is your hourly pay rate?");
        double rate = keyb.nextDouble();

        // Get the hours worked
        System.out.println("How many hours did you work?");
        double hours = keyb.nextDouble();

        // Calculate and display the gross pay
        double gross = rate * hours;
        System.out.println("Your gross pay is "+ gross);
    }
}
Character Data

• All of our programs so far have used variables to store numbers, not words.
• We can store single characters by writing:
  
  ```
  char x, y;
  ```

  – x and y can hold one and only one character

• For now, we use character data for input and output only.

Character Strings

• We are usually interested in manipulating more than one character at a time.
• We can store more than one character by writing:
  
  ```
  String s = new String();
  ```

• If we want s can hold to have some initial value, we can write:
  
  ```
  String s
  = new String("Initial value");
  ```

• For now, we use character data for input and output and concatenation only.
A program that uses a character variable

```java
import java.util.Scanner;

public class Polite {
    // A very polite program that greets you by name
    public static void main(String[] args) {
        String name = new String();
        Scanner keyb = new Scanner(System.in);
        // Ask the user his/her name
        System.out.println("What is your name?");
        name = keyb.next();
        // Greet the user
        System.out.println("Glad to meet you, " + name);
    }
}
```

Using Stepwise Refinement to Design a Program

- You should noticed that when we write a program, we start by describing the steps that our program must perform and we subsequently refine this into a long series of more detailed steps until we are writing individual steps. This is called **stepwise refinement**.
- Stepwise refinement is one of the most basic methods for developing a program.
Example – A program to convert pounds to kilograms

- Our program will convert a weight expressed in pounds into kilograms.
  - Our input is the weight in pounds.
  - Our output is the weight in kilograms
  - We also know that
    Kilograms = Pounds / 2.2

Pounds to Kilograms Program (continued)

- Our program must:
  1. Get the weight in pounds
  2. Calculate the weight in kilograms
  3. Print the weight in kilograms
Pounds to Kilograms Program (continued)

- Our program must:
  1. Get the weight in pounds
  2. Calculate the weight in kilograms
  3. Print the weight in kilograms

  1.1 Prompt the user for the weight in pounds
  1.2 Read the pounds

Pounds to Kilograms Program (continued)

- Our program must:
  1.1 Prompt the user for the weight in pounds
  1.2 Read the pounds
  2. Calculate the weight in kilograms
  3. Print the weight in kilograms

```java
System.out.println
  ("What is the weight in pounds?");
double lbs = keyb.nextInt();
```
Pounds to Kilograms Program (continued)

```java
System.out.println
    ("What is the weight in pounds?");
double lbs = keyb.nextInt();
2. Calculate the weight in kilograms
3. Print the weight in kilograms
    double kg = lbs / 2.2;

System.out.println("The weight is " + kg + " kilograms");
```
import java.util.Scanner;

public class ConvertPounds {

    // Convert pounds to kilograms
    // Input - weight in pounds
    // Output - weight in kilograms
    public static void main(String[] args) {
        Scanner keyb = new Scanner(System.in);

        // Get the weight in pounds
        System.out.println("What is the weight in pounds?");
        double lbs = keyb.nextInt();

        // Calculate and display the weight in kilograms
        double kg = lbs / 2.2;
        System.out.println("The weight is "+kg +" kilograms");
    }
}

Another Example – The Area of A Rectangle

• Our program will calculate the area of a rectangle.
  – Our input is the length and width.
  – Our output is the area.
  – We also know that
    Area = Length * Width
Our Program’s Steps

1. Find the length and width
2. Calculate the area
3. Print the area

Our Program’s Steps (continued)

1. Find the length and width
2. Calculate the area
3. Print the area

1.1 Find the length
1.2 Find the width
Our Program’s Steps (continued)

1. Find the length
2. Calculate the area
3. Print the area

1.1 Find the length
1.2 Find the width

1.1.1 Prompt the user for the length
1.1.2 Read the length
1.2.1 Prompt the user for the width
1.1.2 Read the width

2. Calculate the area
3. Print the area

System.out.println("Enter the length?");
double length = keyb.nextDouble();

System.out.println("Enter the width?");
double width = keyb.nextDouble();
Our Program’s Steps (continued)

```java
System.out.println("Enter the length?");
double length = keyb.nextDouble();

System.out.println("Enter the width?");
double width = keyb.nextDouble();
```

2. Calculate the area
3. Print the area

```
double area = length * width;
```

Our Program’s Steps (continued)

```java
System.out.println("Enter the length?");
double length = keyb.nextDouble();

System.out.println("Enter the width?");
double width = keyb.nextDouble();
```

```
double area = length * width;
```

3. Print the area

```
System.out.println("The area is " + area);
```
public class CalculateArea {
    // Calculates the area of a rectangle
    // Inputs - The length and width of the rectangle
    // Output - The area of the rectangle
    public static void main(String[] args) {
        Scanner keyb = new Scanner(System.in);

        // Print an explanatory message for the user
        System.out.println("Given the width and length of a rectangle");
        System.out.println("this program calculates its area.");

        // Get the inputs
        System.out.println("Enter the length?");
        double length = keyb.nextDouble();

        System.out.println("Enter the width?");
        double width = keyb.nextDouble();

        // Calculate and display the area
        double area = length * width;
        System.out.println("The area is " + area);
    }
}