Extra Credit Assignment

• Posted on homework section of web site
• Due February 24
  – You must send me email by TODAY with all the appropriate details if you are going to participate
  – I will be sending an email to everyone who has signed up later today
    • if you think you are signed up and you don’t get this email, sign up again!

Questionnaire Results

• Thank you to everyone who responded
• Is the pace of the class too fast, too slow, or just right?
  9 Too fast
  1 Sometimes too fast/usually just right
  20 Just right
  1 Varies between too slow and just right
  1 Too slow
  1 Way too slow

Questionnaire Results

• Suggestions for improvement
  – Have sample problems for the quizzes
  – Have the class slides available before class
  – Explain key points in more detail
  – Ask questions in different ways – all the new vocabulary is daunting
  – Fewer example programs during lecture
    • This is from someone who thinks the pace is too slow
    • Make sure the TAs know how to do the timesheet

Questionnaire Results

• A comment from one student:
  "Actually, the class is good, it’s the students who need to ask questions & wake-up."
• For those of you who think I am going to fast, there’s a message here
  – The pace will automatically slow down if you ask questions.

Homework 4 Comments

• Very good overall
• Some common problems
  – Poor choice of variable names
    • a, b, p, t, x are not adequate
    • roll, point are much better names
  – Lines that are too long
    • Hard to read
    • Don’t print well
    • Try to limit to 72 to 80 characters max

Chapter 5: Arrays

• What if you had a bunch of quiz scores that you wanted to store and print?
  – You could have a variable for each one
    int quizScoreCrystal;
    int quizScoreWilliam;
    int quizScoreEric;
    int quizScoreMallory;
  – This is not very simple
    • What if you wanted to sort them?
    • What if you wanted to print them?
Array
- An array allows you to store many values using a single identifier
  `int[] quizScore;`
- Each element of the array can be accessed individually using its index or subscript.
  `quizScore[0]` - the first element
  `quizScore[1]` - the second element
  `quizScore[i]` - the i+1st element

Arrays in memory

Array Declaration
- Use `[]` to declare array variables
  `int[] quizScore;`  - declares an array of ints
  `double[] temperature;`  - declares an array of doubles
  `String[] lastName;`  - declares an array of Strings
- The declaration does not create or initialize the array.
- It declares the array reference only

Declaring the array reference

Create the array
- After you have declared an array variable, you need to create the array
  `arrayVariable = new type [ length ];`
- The `new` operator allocates memory for the array.
- This creates an array of the given type with length elements.
- `arrayVariable` is assigned the reference to newly created array.

Examples
- Create an array reference and an array of 65 ints.
  ```java
  int[] quizScore;  
  quizScore = new int[ 65 ];
  ```
  - This elements of this array will be indexed from 0 to 64.
  - The elements will be initialized with the value 0.
Examples

- Create an array reference and an array of `count` doubles

```java
int count;
double[] temperature;
count = Console.in.readInt();
temperature = new double[ count ];
```

- We won’t know how big this array is going to be until we run the program

Creating the array

```java
int[] quizScore;
quizScore = new int[65];
```

Examples

- You can create the array reference and the array in one statement:

```java
int[] quizScore = new int[ 65 ];
double[] temperature = new double[ count ];
boolean[] isFemale = new boolean[ 100 ];
```

Accessing Array Elements

- Individual array elements are accessed by subscripting or indexing.
- The subscript value ranges from 0 to the array length – 1
- The subscript is an integer expression

Array Elements and Assignment

- Use subscripting to assign a value to an array element:

```java
quizScore[10] = 45;
temperature[13] = 12.6;
```

```java
int[] example = new int[10];
for ( int i = 0; i < 10; i++ ) {
    example[i] = i * 2;
}
Accessing Array Elements

- Use subscripting to access the value of an array element:

```java
System.out.print("quizScore[10] = " + quizScore[10]);
if ( temperature[13] <= 32.0 ) {
    // Do something here
}
for ( int i = 0; i < 10; i++ ) {
    System.out.println(i + " " + example[i]);
}
```

Array index

- The value of the array index expression must be an integer
- The value of the array index must be in the range 0 to array length – 1
  – If you try to access outside this range, Java generates an IndexOutOfBoundsException

```java
int[] example = new int[10];
for ( int i = 0; i <= 10; i++ ) {
    System.out.println(i + " " + example[i]);
}
```

Initializing arrays

- You can create the array reference, allocate the array, and initialize it in one statement

```java
double[] data = {5.1, 3.7, 6.2, 10.3, 4.5};
int[] numDaysPerMonth = {31, 28, 31, 30, 31, 30, 31, 30, 31, 30, 31, 31};
```

Array Initialization

```java
double[] data = {5.1, 3.7, 6.2, 10.3, 4.5};
```

Array length

- An array stores more than just the elements of the array
- It also stores the array's length
- Use `<arrayname>.length` to get this value

```java
int[] example = new int[10];
for ( int i = 0; i < example.length; i++ ) {
    System.out.println(i + " " + example[i]);
}
```

- This style is preferred over one with literals
  – Much easier to change your program if you decide you need to change the array size