

## Week 3, Lecture 2

### Control Constructs



1

## Announcements

- **1st Assignment** is due at the end of next week (deadline **4pm Friday August 20th**).
- to submit your assignment go to the course home page, click on **Assignment Submission** and follow the instructions

2

## Assignment 1 Submission

- prepare the file **ExpandingCircles.java**
- complete the comments at the top of this file, e.g.:
  - // **Family name:** Xmas
  - // **Given names:** Mary Jane
  - // **Student number:** 12345678
- follow the instructions for submission

3

## Announcements (cont)

- if you have any questions about a Quick Quiz problem, ask a tutor in the lab

4

## This Week

**Java Genesis:**

–**Ch4: Control constructs**

**Lab Assessment 3** (deadline **Week 5**)

**Quick Quiz for Ch3**

5



## Problem

Create a program to test if a given +ve integer is a prime number.

More generally, create a program that lists all the prime factors of a given +ve integer.

6

## Specification

Given a +ve integer, list every +ve integer which

- is a prime number;
- divides exactly into the given integer.

7

## Example

Consider the +ve integer **144**:

factors of **144** are

**1, 2, 3, 4, 6, 9, 12, 16, 18, 24, 36, 48, 72, 144**

of which only **2** and **3** are prime.

Hence the prime factors of **144** are **2** and **3**.

8

## A simpler problem?

Create a program that lists **all** the factors of a given +ve integer.

9

## A scenario to extract all factors

Suppose *num* is the given +ve integer

start with *taf* = 1

- (a) if *taf* divides exactly into *num* then *taf* is a factor; add *taf* to the list of factors of *num*
- (b) increase *taf* by 1 and go back to (a), stopping when *taf* > *num*

10

```
import genesis.*;
public class AllFactors {
    public static void main (String [ ] args) {
        int num = DialogBox.requestInt
            ("supply an integer:");
        Transcript.println
            ("The factors of "+num+" are");
        int taf = 1;
        while (taf <= num/2) {
            if (num % taf == 0) {
                Transcript.print(taf + " ");
            }
            taf++;
        }
        Transcript.println(num);
    }
}
```

11

How can we extract just the **prime** factors rather than all the factors?

## An observation:

the smallest factor (other than 1) of a number is always prime.

12

## A scenario to extract prime factors

Suppose  $num$  is the given +ve integer

1. let  $taf = 2$
2. if  $taf$  divides  $num$  then  $taf$  is a prime factor;  
add  $taf$  to the list of prime factors of  $num$
3. replace  $num$  by  $num/taf$  ;  
if  $taf$  still divides  $num$  repeat this step
4. increase  $taf$  by 1 and go back to step 2,  
stopping when  $num$  equals 1

13

## Example

Find the prime factors of **15925**:

**2, 3 and 4 are not factors of 15925**

**5 is a factor of 15925:  $15925 / 5 = 3185$**

**5 is a factor of 3185:  $3185 / 5 = 637$**

**5 and 6 are not factors of 637**

**7 is a factor of 637:  $637 / 7 = 91$**

**7 is a factor of 91:  $91 / 7 = 13$**

**7, 8, 9, 10, 11, and 12 are not factors of 13**

**13 is a factor of 13:  $13 / 13 = 1$**

14

## Another observation:

if a number is not prime, its smallest prime factor is less than or equal to the square root of the number.

15

```
import genesis.*;
public class PrimeFactors {
    public static void main (String [ ] args) {
        int num = DialogBox.requestInt("supply an integer:");
        Transcript.println("Prime factors of "+num+" are");
        int taf = 2;
        while (num != 1) {
            if (num % taf == 0) {
                Transcript.print(taf + " ");
                do num = num / taf;
                while (num % taf == 0);
            }
            taf++;
            if (taf > Math.sqrt(num)) taf = num;
        }
        Transcript.println("Done");
    }
}
```

16

How long will it take to show that a 100 digit number is prime?

Suppose we can check  $10^6$  possible factors in a second. In 1 year we can check about

$$10^6 \cdot 3 \cdot 10^7 < 10^{14}$$

possible factors.

Total time to check the 100 digit number is therefore  $10^{50}/10^{14} = 10^{50-14} = 10^{36}$  years!

17

## PRIMES is in P

Agrawal, Kayal and Saxena

(Indian Institute of Technology, Kanpur)

August, 2002

<http://www.cse.iitk.ac.in/primalty.pdf>

18