Break and Continue and Example Programs

Hui Chen

Department of Computer & Information Science

Brooklyn College

Objectives

- To implement program control with break and continue (§5.9).
- To write a program that displays prime numbers (§5.11).

Outline

- Discussed
    - Loops and While loops
    - Design while loops
    - while Loop vs. do-while Loop vs. for Loop
    - Pitfalls and Errors
    - Nested Loops
    - Algorithms and Example Programs
- Loop control using break and continue
- Algorithms and Example Programs
    - Palindromes
    - Prime numbers

Using break and continue

- break and continue can control the execution of a loop

break

```
public class TestBreak {
  public static void main(String[] args) {
    int sum = 0;
    int number = 0;

    while (number < 20) {
      number++;
      sum += number;
      if (sum >= 100)
        break;
    }

    System.out.println("The number is " + number);
    System.out.println("The sum is " + sum);
  }
}
```

continue

```
public class TestContinue {
  public static void main(String[] args) {
    int sum = 0;
    int number = 0;

    while (number < 20) {
      number++;
      if (number == 10 || number == 11)
        continue;
      sum += number;
    }

    System.out.println("The sum is " + sum);
  }
}
```

Questions?

Guessing Number Problem Revisited

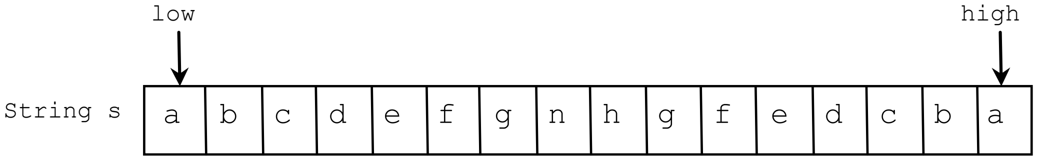
- “break” out of the loop when the guess is correct.

Problem. Checking Palindromes

- A string is a palindrome if it reads the same forward and backward. The words “mom,” “dad,” and “noon,” for instance, are all palindromes.
- The problem is to write a program that prompts the user to enter a string and reports whether the string is a palindrome.

Solution. Checking Palindromes

- Check whether the first character in the string is the same as the last character. If so, check whether the second character is the same as the second-to-last character.
- This process continues until a mismatch is found or all the characters in the string are checked, except for the middle character if the string has an odd number of characters.



Problem. Displaying Prime Numbers

- Problem: Write a program that displays the first 50 prime numbers in five lines, each of which contains 10 numbers. An integer greater than 1 is *prime* if its only positive divisor is 1 or itself. For example, 2, 3, 5, and 7 are prime numbers, but 4, 6, 8, and 9 are not.

Solution. Displaying Prime Numbers

- Solution: The problem can be broken into the following tasks:
  1. For number = 2, 3, 4, 5, 6, ..., test whether the number is prime.
  2. Determine whether a given number is prime.
  3. Count the prime numbers.
  4. Print each prime number, and print 10 numbers per line.

Questions