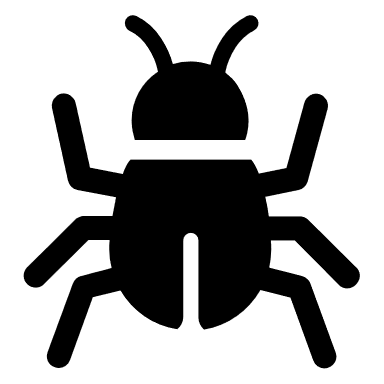
|  |  |
| --- | --- |
| import java.util.Scanner;    public class ReviewSession {     public static void main(String[] args) {         int num;  String color;   Scanner in = new Scanner(System.in);         System.out.print("Pick a whole number: ");         num = in.nextInt();   System.out.print("Pick a color: ");   color = in.nextLine();   System.out.printf("%d %s balloons\n", num,      color);      }  }                             Final Print:                                                                           5 balloons | import java.util.Scanner;    public class ReviewSession {     public static void main(String[] args) {          int num;  String color;  Scanner in = new Scanner(System.in);         System.out.print("Pick a whole number: ");         num = in.nextInt();         in.nextLine();   System.out.print("Pick a color: ");   color = in.nextLine();   System.out.printf("%d %s balloons\n", num,      color);      }                               Final Print:  }                                                                              5 red balloons |



This is a good example of the Scanner bug.

Whenever we read in an int (nextInt()) or a

double (nextDouble()) and want to read in

a String (next() or nextLine()) we need to make sure that we read in the newline character that the scanner receives after the user types in a number (int or double).

The example on the left is incorrect because it does not handle the Scanner bug. The user is quickly prompted but never allowed to type in their response to picking a color because the Scanner reads in the newline character as the user’s response.

The example on the right hand side shows us how we can avoid the Scanner Bug!

|  |  |
| --- | --- |
| public class ReviewSession {     public static void main(String[] args) {  double fraction = 1.0 / 3;  String message = “The value of 1/3 with two” +     “decimal places”;         System.out.print(message + “ “ + fraction);      }  }  Printed:  The value of 1/3 with two decimal places 0.33333333 | public class ReviewSession {  public static void main(String[] args) {  double fraction = 1.0 / 3;  String message = “The value of 1/3 with two” +     “decimal places”;    System.out.printf(“%s %.2f”, message,     fraction);      }  }  Printed:  The value of 1/3 with two decimal places 0.3 |

Using printf allows us to format Strings, and numeric values.

The code on the left is incorrect. Without using printf the value of num will not have two decimal places instead it will have many (ex The value of 1 / 3 with two decimal places 0.3333333333)

The code on the right is correct. It properly using printf to format the value of num (ex. The value of 1 / 3 with two decimal places  0.33)