



Reading Parameters

CSCI 201

Principles of Software Development

Jeffrey Miller, Ph.D.
jeffrey.miller@usc.edu



Outline

- Reading Data from a User
- Program

Hello CSCI 201 Program



```
// Java Hello CSCI 201 Program
```

```
1 public class Test {
2     private int func() {
3         return 201;
4     }
5     public static void main(String [] args) {
6         Test t = new Test();
7         System.out.println ("Hello CSCI " + t.func());
8     }
9 }
```

```
// C++ Hello CSCI 201 Program
```

```
1 #include <iostream>
2 using namespace std;
3 class Test {
4     public:
5         int func() {
6             return 201;
7         }
8 };
9 void main() {
10     Test t;
11     cout << "Hello CSCI " << t.func() << endl;
12 }
```



- Most Java code is inside of a class
- Only one `public` class can exist in a file
- The name of the file must be the same as the name of the public class in the file
- Java is case-sensitive
- The syntax of the main method is
 - › `public static void main(String [] args)`
 - › The `args` variable is used for reading parameters from the command line, different than reading parameters from the user when prompted

Reading Input from a User (Scanner)



```
1  import java.util.Scanner;
2
3  public class ReadInput {
4      public static void main(String [] args) {
5          Scanner in = new Scanner(System.in);
6          System.out.print("Enter your name: ");
7          String name = in.nextLine();
8          System.out.print("Enter the amount: ");
9          double amount = in.nextDouble();
10         in.close();
11         System.out.println("Name = " + name);
12         System.out.println("Amount = " + amount);
13     }
14 }
```

Reading Input from a User (BufferedReader)



```
1  import java.io.BufferedReader;
2  import java.io.InputStreamReader;
3  import java.io.IOException;
4  public class Test {
5      public static void main(String [] args) {
6          InputStreamReader isr = new InputStreamReader(System.in);
7          BufferedReader br = new BufferedReader(isr);
8          try {
9              System.out.print ("Enter a number: ");
10             String line = br.readLine();
11             int num = Integer.parseInt(line);
12             System.out.println ("You entered " + num);
13         } catch (IOException ioe) {
14             System.out.println ("IOException: " + ioe.getMessage());
15         }
16     }
17 }
```

Reading Input from a User (Command Line)

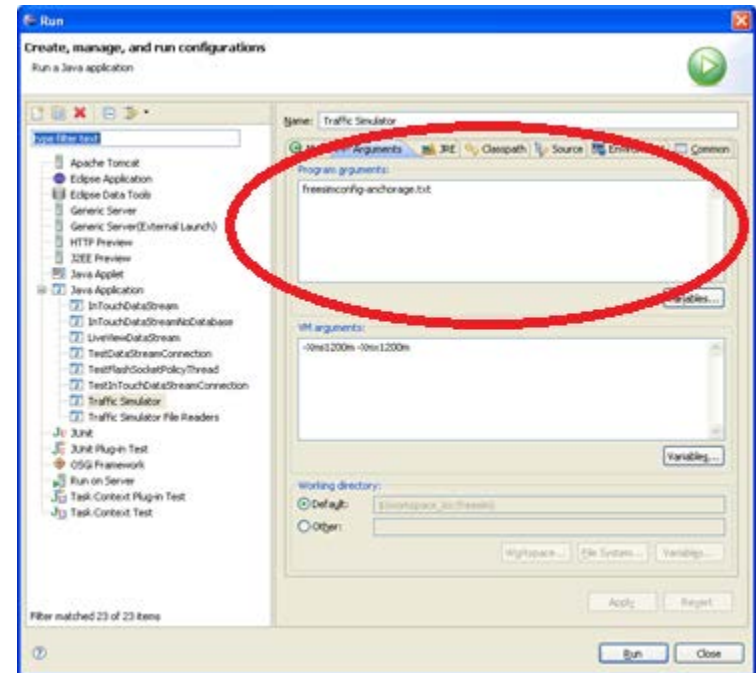


```
1 public class ReadCommandLine {
2     public static void main(String [] args) {
3         if (args.length == 2) {
4             String name = args[0];
5             double amount = Double.parseDouble(args[1]);
6             System.out.println("Name = " + name);
7             System.out.println("Amount = " + amount);
8         }
9     }
10 }
```

Setting Command Line Parameters



- If you are running from the command line, the command line parameters immediately follow the name of the class
 - › `java ReadCommandLine Jeff 2000.30`
- If you are in Eclipse, go to:
 - › Run->Open Run Dialog
 - › Select your application on the left
 - › Click the “Arguments” tab on the right
 - › Command line parameters go in the “Program arguments:” box





Outline

- Reading Data from a User
- Program

Program



- Write a program that reads data from a user that represents employee names and hourly rates. Calculate the annual salary of the employee assuming 50 weeks at 40 hours/week are worked each year. Here is a sample execution with user input bolded:

```
c:\>java com.uscInc.Salary
How many employees? 2
Please enter name of employee #1: Mary
Please enter hourly rate of Mary: 10.50
Please enter name of employee #2: Tony
Please enter hourly rate of Tony: 20.00
Mary's annual salary is $21000.00
Tony's annual salary is $40000.00
c:\>
```