Name: _____ Session: AM / PM
Circle one

Chapter 5 - Day 3

Problem Solving with Methods

FUNCTIONAL DECOMPOSITION

One task per method: Functional decomposition breaks a problem into tasks and gives each task its own method. An input method returns what it read, a calculation method takes values and returns a _____, and an output method is void and takes the values it shows. Then main reads like the steps of the problem.

Term	Definition / memory cue
functional decomposition	
single responsibility	

FD1 One task per method `FD1_FuelConverter.java`

What changed: main converts fuel from gallons to liters with three calls: input, calculation, and output.

```
4  public static void main(String[] args) {
5      Scanner keyboard = new Scanner(System.in);
6      double gallons = getGallons(keyboard);
7      double liters = gallonsToLiters(gallons);
8      displayResults(gallons, liters);
9      keyboard.close();
10 }
```

1. getGallons must return a _____
2. gallonsToLiters needs _____ argument
3. displayResults returns _____

Practice in pairs. Problem: ask for an astronaut's Earth weight, calculate their weight on Mars (Earth weight times 0.38), and show both. Fill in each method's parameters and return type.

Step	Method	Parameters	Returns
Read the Earth weight	getEarthWeight		
Calculate the Mars weight	marsWeight		
Show both weights	displayWeights		

PASSING THROWS UP

The throws clause: A method that opens a file needs `throws IOException` in its header. Any method that calls it needs the clause too, all the way up to _____. A method that only uses the keyboard doesn't need it.

Term	Definition / memory cue
throws clause	

TH1 A method that reads a file

`TH1_FuelLogReport.java`

What changed: getTotalFuel opens the log file and adds up its numbers. main calls it.

```
 6 public static void main(String[] args) throws IOException {
 7     Scanner keyboard = new Scanner(System.in);
 8     String fileName = getFileName(keyboard);
 9     double total = getTotalFuel(fileName);
10     displayResults(total);
11     keyboard.close();
12 }
13
14 public static String getFileName(Scanner keyboard) {
15     System.out.print("Enter the log file name: ");
16     return keyboard.nextLine();
17 }
18
19 public static double getTotalFuel(String fileName)
20         throws IOException {
```

1. getFileName has no clause because it only reads the
2. main needs the clause because it calls

Practice in pairs. Which methods need throws IOException? The first four rows belong to one crew report program.

Method	What it does	Needs it?
`getFileName(Scanner keyboard)`	Reads a file name from the keyboard	
`readCrewTotal(String fileName)`	Opens the file and adds up its numbers	
`main`	Calls getFileName, readCrewTotal, and showReport	
`showReport(double total)`	Prints the total	
Any method	In a program that reads only from the keyboard	

COMMON ERRORS

Fix one, compile again: When javac finds a syntax error, it may stop before it checks the types, so some errors only show up after others are fixed. Fix one error, compile again, and _____ until it compiles.

Method error checklist (fill in the fix as we go through the slides)

Example	Error	Fix
`void showCrew(int count);`	A semicolon after a header	
`void showHeader();`	void written in a call	
`showHeader;`	Missing parentheses	
`showCrew();`	Missing argument for showCrew(int count)	
`(double cargo, extra)`	A parameter with no type	
`showMass(payload)`	A double passed to an int parameter	
`println(thrust) in main`	A parameter used outside its method	
`int x; println(x);`	A local variable with no value	
`int sum(...) with no return`	A missing return statement	
`main calls a file method`	A missing throws clause	

CE1 Three errors, two compiles

`CE1_BrokenLaunch.java`

What changed: This program has three planted errors. javac reports two of them on the first compile.

```
2  public static void main(String[] args) {
3      double payload = 1250.5;
4      void showHeader();
5      showMass(payload);
6      System.out.println(totalMass(payload, 300));
7  }
8
9  public static void showHeader() {
10     System.out.println("Launch report");
11 }
12
13 public static void showMass(int kg) {
14     System.out.println("Payload: " + kg + " kg");
15 }
16
17 public static double totalMass(double cargo, extra) {
18     return cargo + extra;
19 }
```

1. Line 4:

2. Line 17:

3. Found on the second compile, line 5:

CE2 Practice in pairs: find the errors

`CE2_BrokenCrew.java`

What changed: Find all three errors and how to fix each one before it compiles.

```
2  public static void main(String[] args) {
3      int crew = 5;
4      showCrew();
5      System.out.println(doubleCrew(crew));
6  }
7
8  public static void showCrew(int count);
9  {
10     System.out.println("Crew: " + count);
11 }
12
13 public static int doubleCrew(int count) {
14     int result = count * 2;
15 }
```

1. Line 4:

2. Line 8:

3. Lines 13-15: