

Teaching Unit 7: Methods, Classes, and Objects

7.1 Methods and Parameters

Problems

Name: _____ Date: _____

Write your reasoning and Java code in the space below. Code uses java.util imports where needed.

7.1.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
int n = 7;  
int copy = n;  
copy += 3;  
System.out.println(n + ":" + copy);
```

7.1.2 - Write code

Write public static void bump(int[] values, int delta). Add delta to each array element. The array mutation must be visible to the caller.

Teaching Unit 7: Methods, Classes, and Objects

7.2 Return Values

Problems

Name: _____ Date: _____

Write your reasoning and Java code in the space below. Code uses java.util imports where needed.

7.2.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
int a = 3;  
int b = 8;  
System.out.println(Math.max(a, b) + Math.min(a, b));
```

7.2.2 - Write code

Write public static int clamp(int value, int low, int high), assuming low <= high. Return the nearest boundary if value lies outside the interval; otherwise return value.

Teaching Unit 7: Methods, Classes, and Objects

7.3 Classes and Constructors

Problems

Name: _____ Date: _____

Write your reasoning and Java code in the space below. Code uses java.util imports where needed.

7.3.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
int[] a = new int[2];  
int[] b = new int[2];  
a[0] = 6;  
System.out.println(a[0] + ":" + b[0]);
```

7.3.2 - Write code

Write a Counter class with a private int value, a constructor accepting its initial value, increment(), and getValue(). The methods operate on instance state.

Teaching Unit 7: Methods, Classes, and Objects

7.4 Object State and Encapsulation

Problems

Name: _____ Date: _____

Write your reasoning and Java code in the space below. Code uses java.util imports where needed.

7.4.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
ArrayList<Integer> a = new ArrayList<Integer>();  
a.add(5);  
ArrayList<Integer> b = a;  
b = new ArrayList<Integer>();  
System.out.println(a.size() + ":" + b.size());
```

7.4.2 - Write code

Write a Wallet class with private int balance initialized to 0, getBalance(), and deposit(int amount). Ignore nonpositive deposits.