COMPSCI 121: METHODS

SPRING 20

GOALS FOR TODAY'S CLASS

Methods continued

- How methods work
- Variable and method scope
- Method name overloading
- Static fields and methods
- Passing Scanner to Methods
- Importing Java packages

This lecture covers content from zyBooks chapter 8.1 to 8.8.

Recursion will be covered after the Spring break

REVIEW: LOCAL & "GLOBAL" VARIABLES

```
public class Test
{
    private int num1;
    private int num2;

    void method1(){
        int num1;
        ...
    }

    public void setX(int num2){
        this.num2 = num2;
    }
}
```

class or global variables

local variable

In programming, "global" means visible throughout the entire program.
In Java, global means visible anywhere in the class scope.
There are global constants in Java (Math.pi for example)

Declare locally to avoid side effects!

this keyword differentiates local and class variable

REVIEW: VARIABLE SCOPE

```
1  public static void swap(int val1, int val2) {
2      int temp;
3      temp = val1;
4      val1 = val2;
5      val2 = temp;
6  }
7
8  public static void main(String[] args) {
9      int x;
10     int y;
11     x = 10;
12     y = 20;
13     swap(x, y);
14     System.out.println(x + ", " + y);
15 }
```

Go to line 14

A variable declared in a method has scope limited to inside that method.

Start line 8

Go to line 1

The output is 10, 20

For a method with a void return type, the method automatically returns execution upon reaching the end of the method's statements.

Clicker Question 1

```
public static void swap(int val1, int val2){
    int temp;
    temp = val1;
    val1 = val2;
    val2 = temp;
}
public static void main(String[] args) {
    int x;
    int y;
    x = 10;
    y = 20;
    swap(x, y);// calls swap method
    System.out.println(x + ", " + y);
}
```

What is the output?

a. 10, 20

b. 20, 20

c. 10, 10

d. 20, 10

Clicker Question 1 Answer

```
public static void swap(int val1, int val2){
    int temp;
    temp = val1;
    val1 = val2;
    val2 = temp;
}
public static void main(String[] args) {
    int x;
    int y;
    x = 10;
    y = 20;
    swap(x, y);// calls swap method
    System.out.println(x + ", " + y);
}
```

Local variable.
Values of x, y in main do not change.

What is the output?

a. 10, 20
b. 20, 20
c. 10, 10
d. 20, 10

Clicker Question 2

```
A. public int addNumbers(int a, int b) {
       int s;
       s = a + b; }
B. public int addNumbers (int a, int b) {
     int s;
     s = a + b;
     return true;}
C. public int addNumbers (int a, int b) {
     s = a + b;
     return s;}
D. public int addNumbers (int a, int b) {
     int s;
     s = a + b;
     return s; }
```

The correct method for adding two numbers and returning the value is

A.

B.

C.

D.

Clicker Question 2 Answer

```
A. public int addNumbers(int a, int b) {
       int s;
       s = a + b; }
B. public int addNumbers (int a, int b) {
     int s;
     s = a + b;
     return true;}
C. public int addNumbers (int a, int b) {
     s = a + b;
     return s;}
D. public int addNumbers (int a, int b) {
     int s;
     s = a + b;
     return s; }
```

The correct method for adding two numbers and returning the value is

A. no return

B. return not int

C. s not initialized

D. Correct

REVIEW: METHOD SCOPE

A method's *scope*, extends from the opening brace to the ending brace of the class.

A method can access any other method defined in the same class, regardless of the order in which the methods are defined.

A method declared as "`public`" allows the programmer to write code that accesses private fields from within a different class.

Clicker Question 3

```
import java.util.*;
public class test{
   public static void shift(int[] nums) {
      int i;
      for(i = 0; i < nums.length-1; ++i) {
         nums[i] = nums[i+1];}
   }
   public static void main(String[] args) {
      int[] ages = {16, 19, 24, 17};
      shift(ages);
      System.out.println(Arrays.toString(ages));
  }
}//end class
```

The main method

A. does not compile.
B. cannot print the changed values of the `age` array.
C. does not have access to the changed values of `ages` in the void `shift` method.
D. compiles and prints the changed values of the `ages` array.

Clicker Question 3 Answer

```
import java.util.*;
public class test{
   public static void shift(int[] nums) {
      int i;
      for(i = 0; i < nums.length-1; ++i) {
         nums[i] = nums[i+1];}
   }
   public static void main(String[] args) {
      int[] ages = {16, 19, 24, 17};
      shift(ages);
System.out.println(Arrays.toString(ages));
  }
}//end class
```

Reference to `ages` array is passed in and values can be changed in the array.

Passes array reference as parameter

How to print array values without using a for-loop.
Needs `java.util` class import.

The main method

A. does not compile.
B. cannot print the changed values of the `age` array.
C. does not have access to the changed values of `ages` in the void `shift` method.
D. compiles and prints the changed values of the `ages` array.

METHOD NAME OVERLOADING

int, int, int

```
public class DatePrinter {
    public void datePrint(int currDay, int currMonth, int currYear) {

        System.out.print(currMonth + "/" + currDay + "/" + currYear);
    }

    public void datePrint(int currDay, String currMonth, int currYear) {

        System.out.print(currMonth + " " + currDay + ", " + currYear);
    }

    public static void main(String[] args) {
        DatePrinter dailyPlanner = new DatePrinter();

        dailyPlanner.datePrint(30, 7, 2012);
        System.out.println();

        dailyPlanner.datePrint(30, "July", 2012);
        System.out.println();
    }
}
```

int, String, int

Invoke `datePrint` method

Clicker Question 4

```java
class Overload {
    int x;
    int y;
    void add(int a) {
        x = a + 1;
        System.out.println(x);
    }
    void add(int a, int b){
        x = a + b;
        System.out.println(x);
    }
}
```

```java
class Overload_main {
    public static void main(String args[])
    {
        Overload obj = new Overload();
        int a = 0;
        obj.add(4);
        obj.add(6, 2);
```

What is the output?

A. 5
   8

B. 6
   8

C. 4
   6
   2

D. 5
   62

Clicker Question 4 Answer

```
class Overload {
    int x;
    int y;
    void add(int a) {
        x = a + 1;
        System.out.println(x);
    }
    void add(int a, int b){
        x = a + b;
        System.out.println(x);
    }
}
```

```
class Overload_main {
    public static void main(String args[])
    {
        Overload obj = new Overload();
        int a = 0;
        obj.add(4);
        obj.add(6, 2);
```

What is the output?

A. 5
   8

B. 6
   8

C. 4
   6
   2

D. 5
   62

STATIC FIELDS & METHODS

```java
public class Store {
    // Declare and initialize private static field
    private static int nextId = 101;

    // Define private fields
    private String name;
    private String type;
    private int id;

    public Store(String storeName, String storeType) {
        name = storeName;
        type = storeType;
        id = nextId;

        ++nextId;   // Increment each time a Store object is created
    }

    public int getId() {
        return id;
    }

    public static int getNextId() {
        return nextId;
    }
}
```

static variable

instance variable

public method for static variable

USING STATIC FIELDS & METHODS

```
public class StoreTester {
    public static void main(String[] args) {
        Store store1 = new Store("Macy's", "Department");
        Store store2 = new Store("Albertsons", "Grocery");
        Store store3 = new Store("Ace", "Hardware");

        System.out.println("Store 1's ID: " + store1.getId());
        System.out.println("Store 2's ID: " + store2.getId());
        System.out.println("Store 3's ID: " + store3.getId());
        System.out.println("Next ID: " + Store.getNextId());
    }
}
```

Calling public methods for instance and static variables

Clicker Question 5

Which statement is correct?

Static variables

1. can be accessed only by private methods.
2. can be accessed only by creating a class object.
3. can be accessed outside the class without using the class name.
4. can be accessed or mutated by a `public static` method from outside of the class.

A. 1, 2, 3, 4
B. 2, 3, 4
C. 3
D. 4

Clicker Question 5 Answer

Which statement is correct?

Static variables

1. can be accessed only by private methods.
2. can be accessed only by creating a class object.
3. can be accessed outside the class without using the class name.
4. can be accessed or mutated by a `public static` method from outside of the class.

A. 1, 2, 3, 4
B. 2, 3, 4
C. 3
D. 4

USING SCANNER IN METHODS

Note: If a method needs to read user input, using multiple Scanners for the same input stream may lead to unexpected results.

```
public static int readNum() {
   Scanner scnr = new Scanner(System.in);
   return scnr.nextInt();}
public static void main(String [] args) {
   int sum = readNum() + readNum() + readNum();
}
```

USING SCANNER IN METHODS

Good practice: create a single Scanner object in main() and use that Scanner object in another method!

```
public static int readNum(Scanner scnr) {
    return scnr.nextInt();}
public static void main(String [] args) {
    Scanner scnr = new Scanner(System.in);
    int sum = readNum(scnr) + readNum(scnr) + readNum(scnr);
}
```

You could use a for-loop to invoke `readNum` with the `scnr` argument!

REMINDER: JAVA PACKAGES

Package	Sample package members	Description
java.lang	String, Integer, Double, Math	Contains fundamental Java classes. Automatically imported by Java.
java.util	Collection, ArrayList, LinkedList, Scanner	Contains the Java collections framework classes and miscellaneous utility classes.
java.io	File, InputStream, OutputStream	Contains classes for system input and output.

```
import java.util.Scanner;
OR
import java.util.*;
```

Two ways to write import statements.

WEEK 8 TO-DO LIST

- Start zyBook chapter 8 exercises.
- Check your iClicker grades in Moodle.
- Communicate with us using only Moodle forum or Piazza.