AP COMPUTER SCIENCE

UNIT #5

WARMUP PROBLEMS

Inheritance / Polymorphism

Warmup

MONDAY 2/29

Happy Leap Day :)

Parrot (parent class)
//age
//sounds

- steal Soul method
- get age method
↳ from Parrot?
- ~~train~~
↳ random √
- ~~speak~~
- constructor

WARMUP

MONDAY
2/29

AP Free Response question

00:00 00

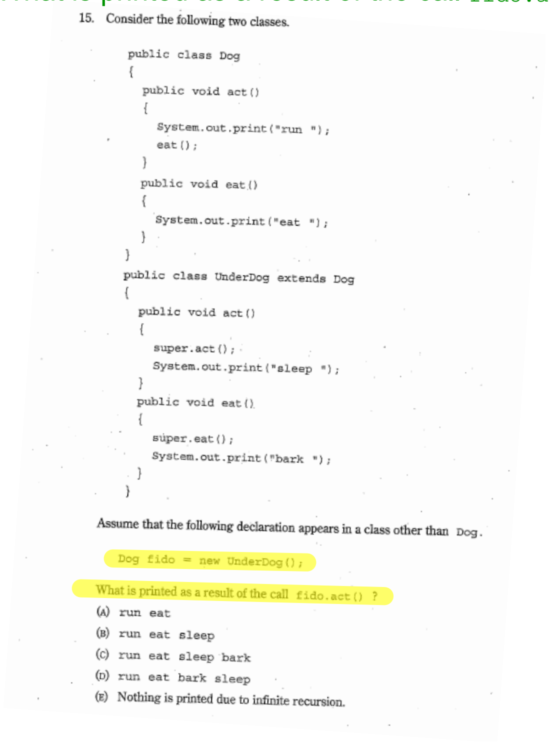
```
public class PirateParrot extends Parrot {
    //number of souls variable
    private int yearsStolen = 0;
    //constructor method
        //super --> (calls parent class) name
    public PirateParrot(String name) {
        super(name);
        train("Polly want a cracker");   }
    //get age method - overriding
    public int getAge()  {
        return super.getAge( ) + yearsStolen;
    //steal souls method
    public void stealSoul (int soulAge)  {
        yearsStolen += soulAge;  }
```

Parrot (parent class)
//age
//sounds

WARMUP

TUESDAY
3/1

What is printed as a result of the call `fido.act()` ?



15. Consider the following two classes.

```
public class Dog
{
  public void act()
  {
    System.out.print("run ");
    eat();
  }
  public void eat()
  {
    System.out.print("eat ");
  }
}
public class UnderDog extends Dog
{
  public void act()
  {
    super.act();
    System.out.print("sleep ");
  }
  public void eat()
  {
    super.eat();
    System.out.print("bark ");
  }
}
```

Assume that the following declaration appears in a class other than `Dog`.

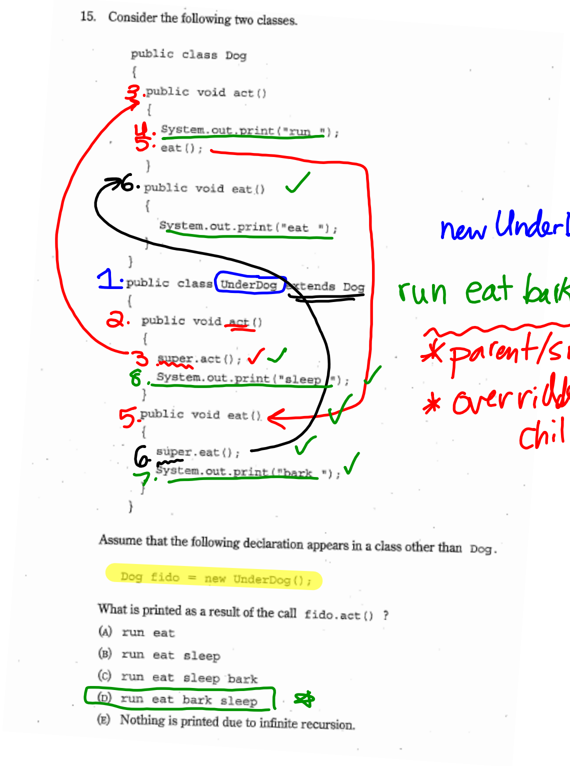
`Dog fido = new UnderDog();`

What is printed as a result of the call `fido.act()` ?

(A) `run eat`
(B) `run eat sleep`
(C) `run eat sleep bark`
(D) `run eat bark sleep`
(E) Nothing is printed due to infinite recursion.

answers...

TUESDAY 3/1



15. Consider the following two classes.

```
    public class Dog
    {
3.    public void act()
      {
4.      System.out.print("run ");
5.      eat();
      }
6.    public void eat()            ✓
      {
        System.out.print("eat ");
      }
    }
1:  public class UnderDog extends Dog
    {
2.    public void act()
      {
3       super.act();  ✓✓
8.      System.out.print("sleep ");  ✓
      }
5.    public void eat()  ✓
      {
6       super.eat();  ✓
7.      System.out.print("bark ");  ✓
      }
    }
```

new UnderDog

run eat bark sleep

*parent/super 1st

* overridden by child

Assume that the following declaration appears in a class other than `Dog`.

```
Dog fido = new UnderDog();
```

What is printed as a result of the call `fido.act()` ?

(A) run eat

(B) run eat sleep

(C) run eat sleep bark

[✓ (D) run eat bark sleep]

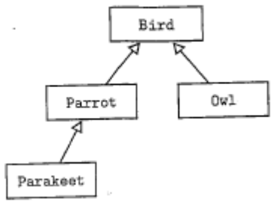
(E) Nothing is printed due to infinite recursion.

WARMUP

WEDNESDAY 3/2

Bird inheritance Multiple Choice Problem

27. Consider the following hierarchy of classes:



Assuming that each class has a valid default constructor, which of the followi declarations in a client program are correct?

```
I   Bird b1 = new Parrot();
    Bird b2 = new Parakeet();
    Bird b3 = new Owl();

II  Parakeet p = new Parrot();
    Owl o = new Bird();

III Parakeet p = new Bird();
```

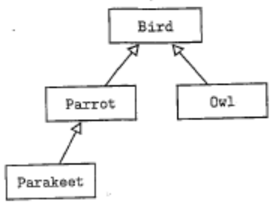
(A) I only
(B) II only
(C) III only
(D) II and III only
(E) I, II, and III

answers...

WEDNESDAY 3/2

Bird inheritance Multiple Choice Problem

27. Consider the following hierarchy of classes:



Assuming that each class has a valid default constructor, which of the followi declarations in a client program are correct?

```
I  Bird b1 = new Parrot();
   Bird b2 = new Parakeet();
   Bird b3 = new Owl();
```

```
II ~~Parakeet p = new Parrot();~~
   ~~Owl o = new Bird();~~
```

```
III ~~Parakeet p = new Bird();~~
```

[✓ (A) I only]
(B) II only
(C) III only
(D) II and III only
(E) I, II, and III

Warmup

THURSDAY 3/3

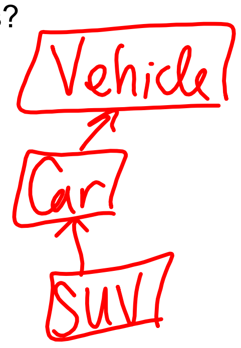
Consider the following classes:

```
public class Vehicle { ... }

public class Car extends Vehicle { ... }

public class SUV extends Car { ... }
```

} draw diagram

Which of the following are legal statements?

```
Vehicle v = new Car();

Vehicle V = new SUV();

Car c = new SUV();

SUV s = new SUV();

SUV s = new Car();

Car c = new Vehicle();
```



answers...

THURSDAY 3/3

Consider the following classes:

```
public class Vehicle { ... }
public class Car extends Vehicle { ... }
public class SUV extends Car { ... }
```

Which of the following are legal statements?

```
Vehicle v = new Car();
Vehicle V = new SUV();
Car c = new SUV();
SUV s = new SUV();
```

~~`SUV s = new Car();`~~

~~`Car c = new Vehicle();`~~

Warmup

FRIDAY 3/4

BJP Section 9.3 - page 618 #9

pg619

```
public class Flute extends Blue {
  public void method 2( ) {
        System.out.println("flute 2");
  }

  public String toString( ) {
        return "flute";
  }
}
```

```
public class Blue extends Moo {
  public void method 1( ) {
        System.out.println("blue 1");
  }
}
```

```
public class Shoe extends Flute {
  public void method1( ) {
        System.out.println("shoe 1");
  }
}
```

```
public class Moo {
  public void method 1( ) {
        System.out.println("moo 1");
  }

  public void method 2( ) {
        System.out.println("moo 2");
  }

  public String toString( ) {
        return "moo";
  }
}
```

WARMUP

MONDAY 3/7

The difference between the "this" keyword and the "super" keyword...

answers...

MONDAY 3/7

The "this" keyword refers to the current object, while the "super" keyword refers to the current class's superclass. Use the "super" keyword when you call a method of constructor from the superclass that you've overridden, and use the "this" keyword when you access your object's other fields, constructors, and methods.

WARMUP

TUESDAY 3/8

LoudDog AP Free Response Question

answers...

TUESDAY 3/8

LoudDog AP Free Response Question

a)

BASE CAT CLASS ON DOG:

```
public class Dog extends Pet
{
    public Dog(String petName)
    { ... }

    public String speak()
    { ... }
}
//the test will give you hints!!
```

```
public class Cat extends Pet
{
    public Cat(String petName)
    {
        super(petName);
    }
    public String speak()
    {
        return "meow";
    }
}
```

b) LoudDog AP Free Response Question

BASE LOUDDOG ON DOG!

```
public class Dog extends Pet
{
    public Dog(String petName)
    { ... }

    public String speak()
    { ... }
}
```

```
public class LoudDog extends Dog
{
    public LoudDog(String petName)
    {
        super(petName);
    }
    public String speak()
    {
        return super.speak() + " " +
                super.speak();
    }
}
```

c) LoudDog AP Free Response Question

```
private List<Pet> petList;

public void allSpeak()
{
    for(Pet p : petList)
    {
        System.out.println(p.getName() + " " + p.speak());
    }

    for(int i = 0; i < petList.size(); i++)
    {
        System.out.println(petList.get(i).getName() + " " +
                                petList.get(i).speak());
    }
```

$$\begin{tikzcd}\boxed{\text{List}} \\ \boxed{\text{ArrayList}} \arrow[u]\end{tikzcd}$$

WARMUP

WEDNESDAY
3/9

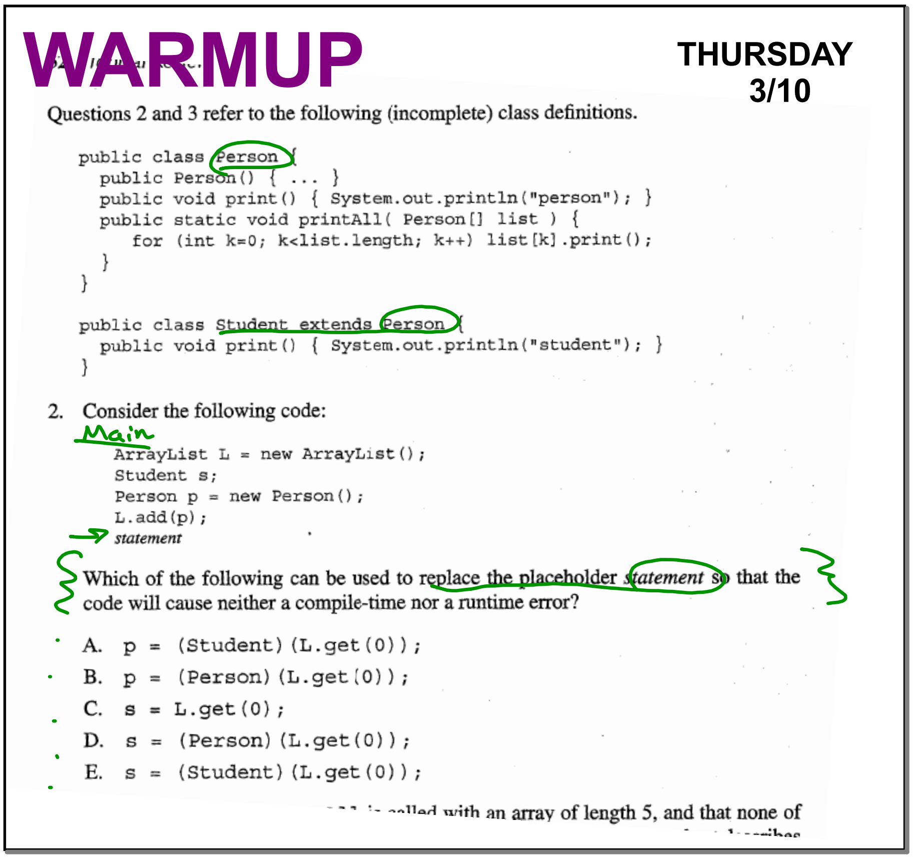
What is the difference between overloading and overriding a method?

answers...

WEDNESDAY
3/9

Overloading a method involves creating two methods in the same class that have the same name but different parameters.

Overriding a method involves creating a new version of an inherited method in a subclass that has identical parameters but new behavior to replace the old.



WARMUP

THURSDAY 3/10

Questions 2 and 3 refer to the following (incomplete) class definitions.

```
public class Person {
  public Person() { ... }
  public void print() { System.out.println("person"); }
  public static void printAll( Person[] list ) {
     for (int k=0; k<list.length; k++) list[k].print();
  }
}

public class Student extends Person {
  public void print() { System.out.println("student"); }
}
```

2. Consider the following code:

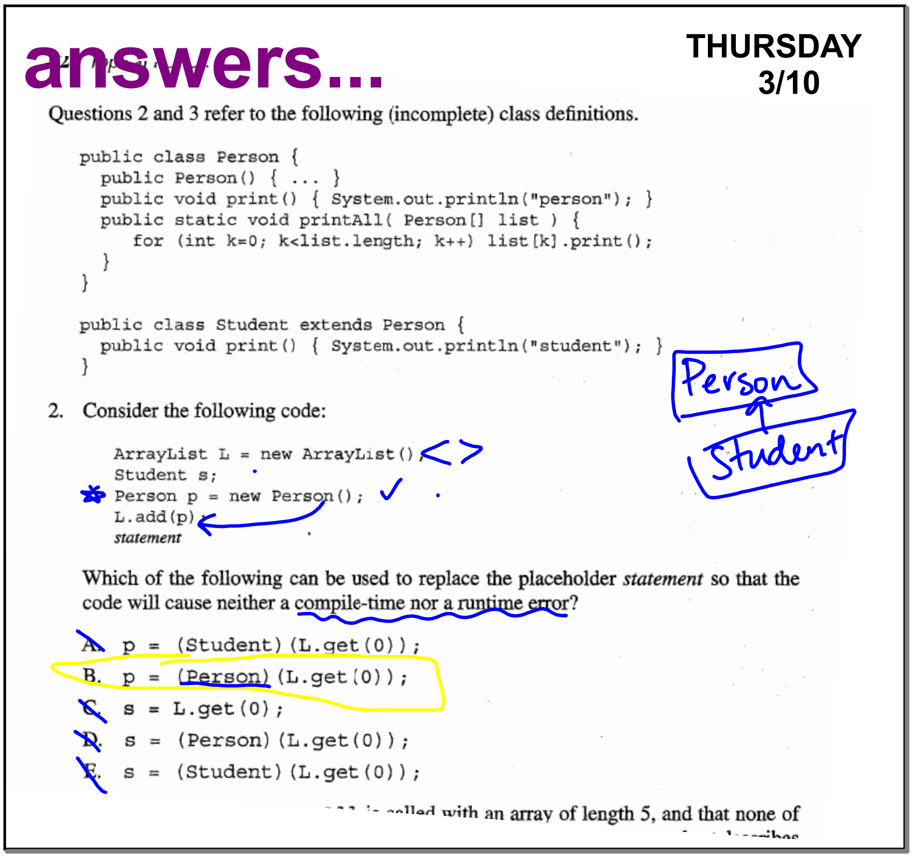
Main

```
    ArrayList L = new ArrayList();
    Student s;
    Person p = new Person();
    L.add(p);
    statement
```

Which of the following can be used to replace the placeholder *statement* so that the code will cause neither a compile-time nor a runtime error?

A. `p = (Student)(L.get(0));`
B. `p = (Person)(L.get(0));`
C. `s = L.get(0);`
D. `s = (Person)(L.get(0));`
E. `s = (Student)(L.get(0));`

called with an array of length 5, and that none of



answers...

THURSDAY 3/10

Questions 2 and 3 refer to the following (incomplete) class definitions.

```
public class Person {
  public Person() { ... }
  public void print() { System.out.println("person"); }
  public static void printAll( Person[] list ) {
     for (int k=0; k<list.length; k++) list[k].print();
  }
}

public class Student extends Person {
  public void print() { System.out.println("student"); }
}
```

Person
↑
Student

2. Consider the following code:

```
    ArrayList L = new ArrayList();<>
    Student s;
  * Person p = new Person(); ✓
    L.add(p)
    statement
```

Which of the following can be used to replace the placeholder *statement* so that the code will cause neither a compile-time nor a runtime error?

~~A.~~ `p = (Student)(L.get(0));`
[✓ B. `p = (Person)(L.get(0));`]
~~C.~~ `s = L.get(0);`
~~D.~~ `s = (Person)(L.get(0));`
~~E.~~ `s = (Student)(L.get(0));`

called with an array of length 5, and that none of

WARMUP

FRIDAY
3/11

AP Exceptions Worksheet

(trace the code, find the output)

answers....

FRIDAY
3/11

AP Exceptions Worksheet

(trace the code, find the output)

thaws thews throws bye

WARMUP

MONDAY 3/14

Go to the following website: http://chortle.ccsu.edu/Java5/

Take these quizzes while I walk around and check off ***Reading Guide- Inheritance #2*** AND ***Lab 13 questions***

Part 6: Object Oriented Programming

~take quizzes, do the review pages and exercises

~do as many as you can during the next 20 minutes

WARMUP

TUESDAY 3/15

BJP Section 9.5 - page 622 #20, 21

20. Consider the following interface and class:

```
public interface I {
     public void m1();
     public void m2();
}
public class C implements I {
   // code for class C
}
```

What must be true about the code for class C in order for that code to compile successfully?

21. What's wrong with the code for the following interface? What should be changed to make a valid interface for objects that have colors?

```
public interface Colored {
     private Color color;
     public Color getColor() {
          return color;
     }
}
```

answers....

TUESDAY 3/15

20. The code for class C must contain implementations of the methods m1 and m2 to compile correctly, because C claims to implement the I interface.

21. The interface is incorrect because interfaces can't declare fields or write bodies for methods. The following code is a correct interface:
import java.awt.*;

```
// Represents items that have a color
//that can be retrieved.

public interface Colored {
     public Color getColor();
}
```

WARMUP

No laptops

WEDNESDAY 3/16

```
//A general interface for shape classes
public interface Shape {
      public double getArea();
      public double getPerimeter();
}
```

Math.Pi

Write the Δ & ○ class.

rectangle class in notes

WARMUP

THURSDAY 3/17

JSS Chapter 7 - page 460, AP Response Multiple Choice #6

20. Consider the following code:

```
public interface Speaker {
     public void speak();
}
public interface Writer {
     public void write();
}
public class Philosopher implements Speaker, Writer {
   //implementation not shown
}
```

Which of the following will NOT cause an error?
A.) `Speaker s = new Philosopher();`
B.) `Speaker s = new Writer();`
C.) `Philosopher p = new Speaker();`
D.) `Philosopher p = new Writer();`
E.) `Object o = new Writer();`

What other combinations of instantiations WILL work?

answers...

THURSDAY 3/17

JSS Chapter 7 - page 460, AP Response Multiple Choice #6

20. Consider the following code:

```
public interface Speaker {
     public void speak();
}
public interface Writer {
     public void write();
}
public class Philosopher implements Speaker, Writer {
   //implementation not shown
}
```

Which of the following will NOT cause an error?
A.) `Speaker s = new Philosopher();`
B.) `Speaker s = new Writer();`
C.) `Philosopher p = new Speaker();`
D.) `Philosopher p = new Writer();`
E.) `Object o = new Writer();`

What other combinations of instantiations WILL work?

WARMUP

FRIDAY
3/18

AP CS WS - Inheritance (Greenlee)

Given the interfaces below answer:

```
public interface Animal
{
   void eat( );
   void sleep( );
   void run( );
}
```

```
public interface Canine
{
   void growl( );
}
```

1) Write the heading for a class called Dog that realizes Animal

2) How many methods must the Dog class have in it if it realizes Animal?

3) Write the heading for a class called Dog that realizes both Animal and Canine

4) How many methods must the Dog class have in it if it realizes Animal and Canine?

Given the interfaces below answer:

```
public interface Animal
{
   void eat( );
   void sleep( );
   void run( );
}
```

```
public interface Canine
{
   void growl( );
}
```

1) Write the heading for a class called Dog that realizes Animal

```
public class Dog implements Animal
```

2) How many methods must the Dog class have in it if it realizes Animal?

```
3, eat(), sleep(), run()
```

3) Write the heading for a class called Dog that realizes both Animal and Canine

```
public class Dog implements Animal, Canine
```

4) How many methods must the Dog class have in it if it realizes Animal and Canine?

```
4, eat(), sleep(), run(), growl()
```

Elevens Activity → make a BlueJ folder

Cards

~~Activity #3~~

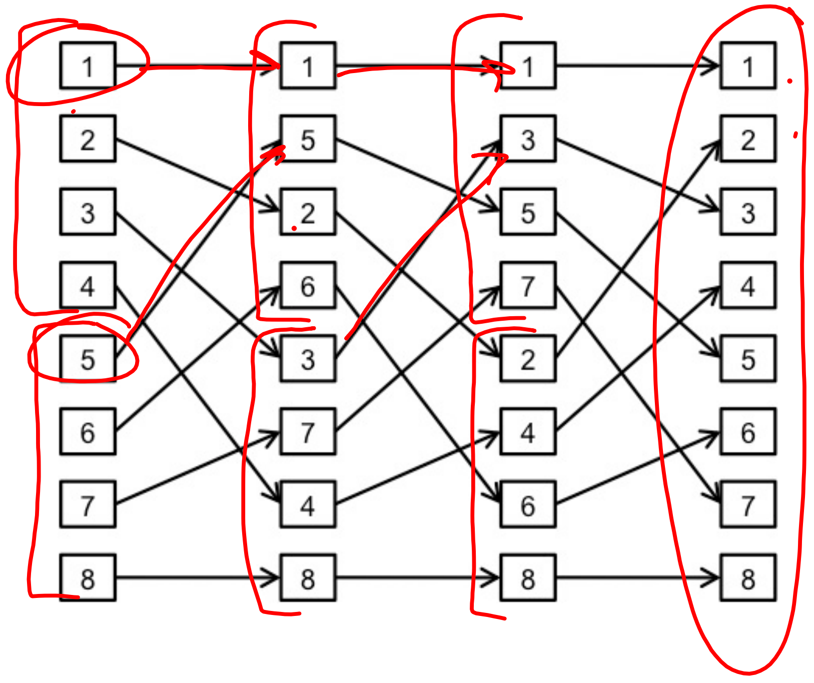
~~Shuffling~~

~~(perfect)~~

#4 → need Activity #3 Solution Code

Shuffling

(random)



$\boxed{\text{Unit 6}}$

Search / Sort / Recursion

```
public class SnowyOwl extends Owl { //1 point
   //Constructor
   public SnowyOwl() { //1 point
         super("Snowy Owl");  //1 point
   }
   public String getFood() { //1 point
      int num = (int) (Math.random() * 3); //1 point
      if(num == 0)    return "hare";
      else if(num == 1) return "lemming";
      else    return "small bird";
}  //1 point - flexible for method
```

WARMUP

MONDAY
3/22

JSS - Chapter 7, AP-Style MC problems - page 458, #1-5

answers...

MONDAY
3/22

2007

JSS - Chapter 7, AP-Style MC problems - page 458, #1-5

1. B 2. D 3. C 4. C 5. C

WARMUP

TUESDAY 3/22

What value is returned by the method call `sum(5)`?

```
public int sum(int n)
{
    if (n == 1)
      return 1;
    else
      return n + sum(n - 1);
}
```

answers...

TUESDAY 3/22

What value is returned by the method call `sum(5)`?

```
public int sum(int n)
{
    if (n == 1)
      return 1;
    else
      return n + sum(n - 1);
}
```

~~5~~ ~~4~~ ~~3~~ ~~2~~ 1

$\text{sum}(5) = 5 + \overset{10}{\cancel{\text{sum}(4)}}$

$\overset{10}{\cancel{\text{sum}(4)}} = 4 + \overset{6}{\cancel{\text{sum}(3)}}$

$\overset{6}{\cancel{\text{sum}(3)}} = 3 + \overset{3}{\cancel{\text{sum}(2)}}$

$\overset{3}{\cancel{\text{sum}(2)}} = 2 + \overset{1}{\text{sum}(1)}$

$\text{sum}(1) = 1$

~~5~~ ~~4~~ ~~3~~ 2   3-1   ~~4-1~~

sum(5) = 15

WARMUP

WEDNESDAY 3/23

```
int result = identity(10);
System.out.println("The final answer is " + result);

public int identity(int num){
  if(num < 1){
    return 10;
  }else{
    return num + identity(num - 2);
  }
}
```

WARMUP

For the following method, what would be displayed by the call: `mystery8("mi-mi-mi")`?

```
public void mystery8(String sWord){
  int nL = sWord.length();
  if(nL >= 3)
  {
 1 mystery8(sWord.substring(0,nL/3));
 2 System.out.println(sWord.substring(nL/3,
                                 2*nL/3));
 3 mystery8(sWord.substring(2*nL/3));
//substring(x) same as substring(x,length())
  }
}
```

Output:

```
-mi
m
```

WARMUP

FRIDAY
3/25

What does the following code in the main print out?

```
int result2 = negative(-3);
System.out.println("The final answer is " + result2);

public int negative(int num)
{
        if(num >= 20){
            return -5;
        }else{
            return negative(num + 4) + 2 * num;
        }
}
```

answers...

FRIDAY
3/25

What does the following code in the main print out?

The final answer is 79

```
int result2 = negative(-3);
System.out.println("The final answer is " + result2);

public int negative(int num)
{
        if(num >= 20){
            return -5;
        }else{
            return negative(num + 4) + 2 * num;
        }
}
```

$neg(-3) = \underbrace{neg(1)} + 2 \cdot -3;$

$neg(1) = neg(5) + 2 \cdot 1;$

END OF UNIT #5