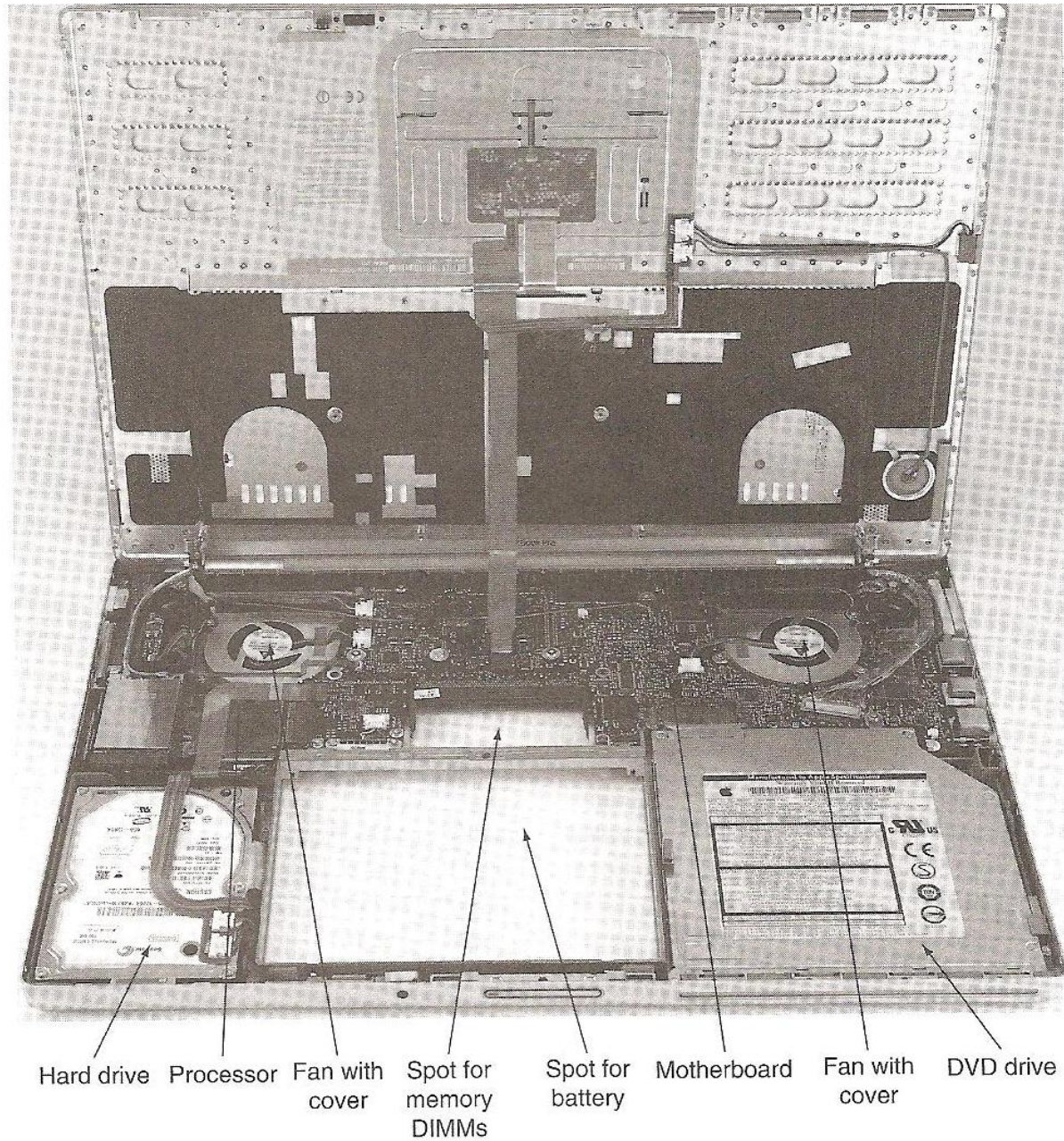


# CS 302: Introduction to Programming

## Chapter 1

# Parts of a Computer



# Parts of a Computer

**Definition: physical computer and peripheral devices**

- **Physical Computer**

- **Motherboard**

- **CPU**

- **Storage Devices**

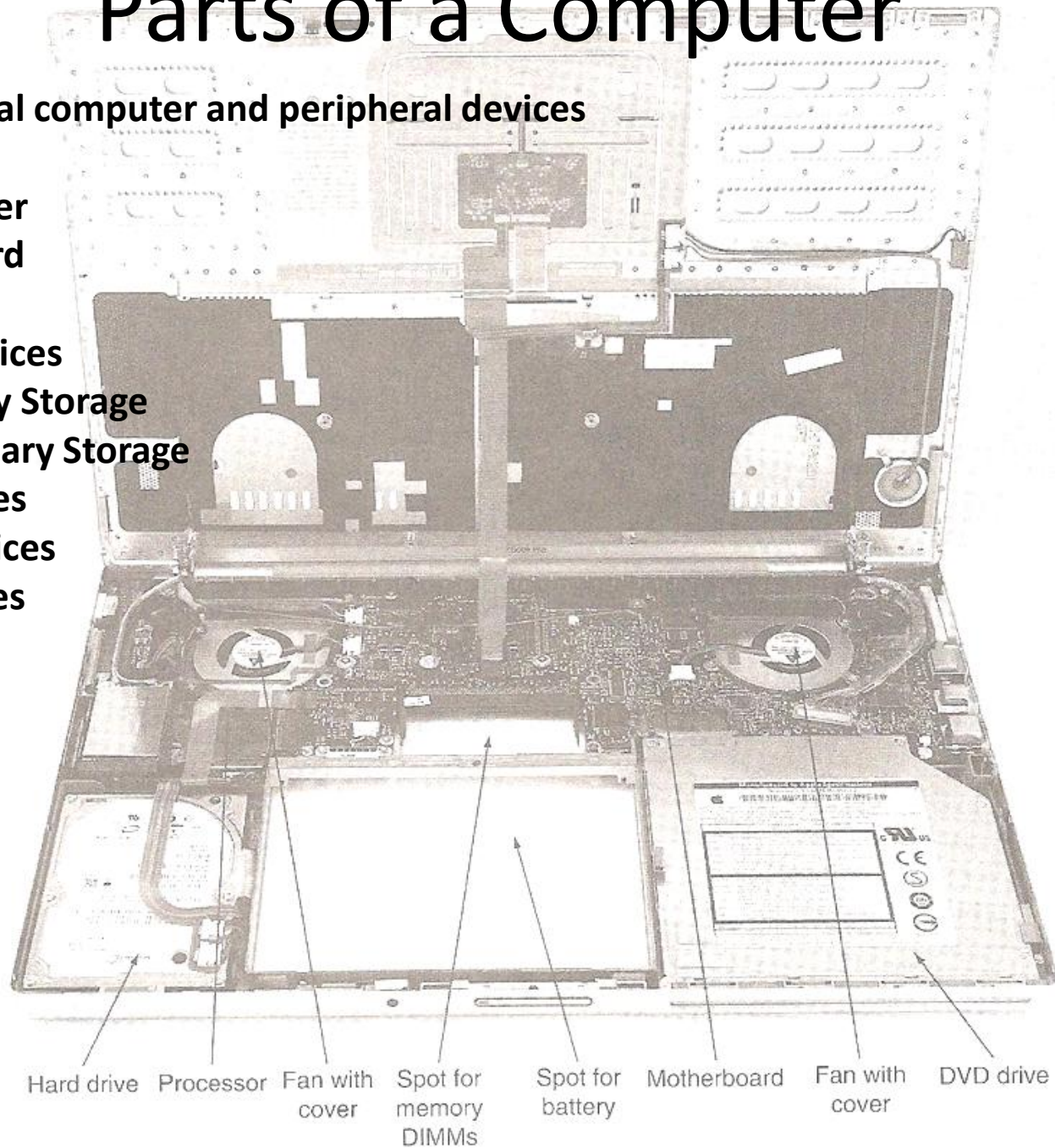
- **Primary Storage**

- **Secondary Storage**

- **Peripheral Devices**

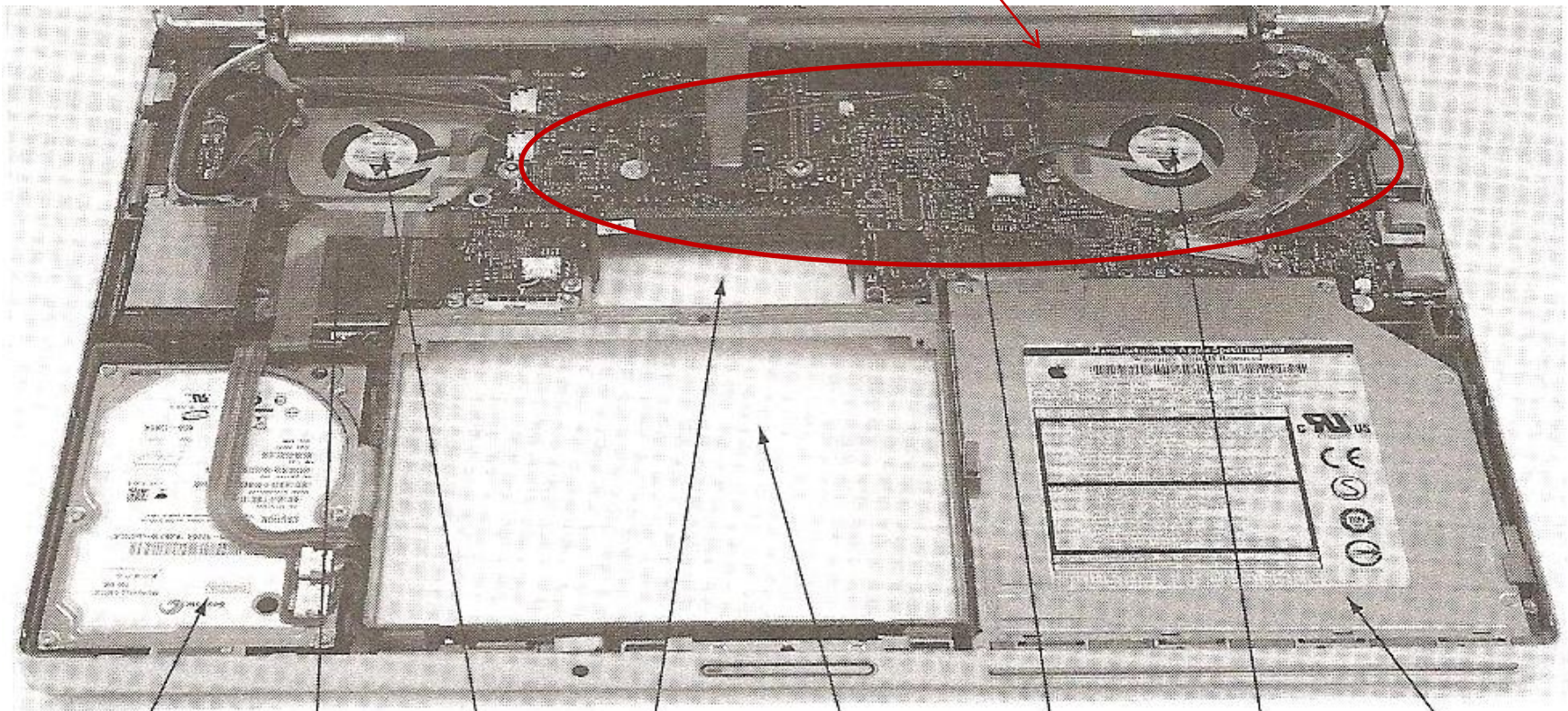
- **Output Devices**

- **Input Devices**



## Motherboard

- The plastic board containing
  - packages of integrated circuits
  - CPU
  - memory
  - I/O connection points.



Hard drive

Processor

Fan with  
cover

Spot for  
memory  
DIMMs

Spot for  
battery

Motherboard

Fan with  
cover

DVD drive  
cover

# CPU: Central Processing Unit

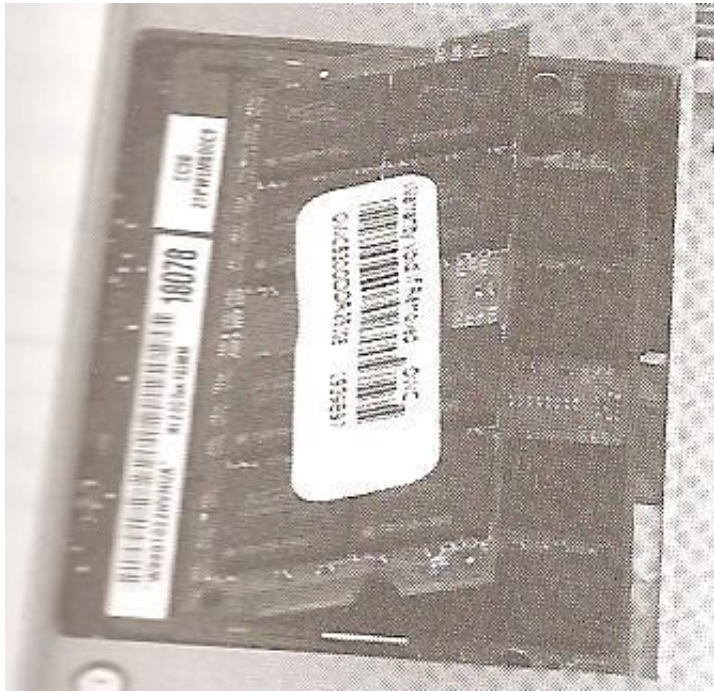
- Locates and Executes Program Instructions
  - Processes Data
  - Fetches data from external memory
  - Stores data to external memory
  - Signals I/O devices to activate
  
- Contains a small cache memory built of SRAM
  
- Parts of the CPU
  - Datapath
    - Performs arithmetic operations
  - Control
    - Commands datapath, memory, and I/O devices

# Storage in the Computer

- Primary Storage (memory)
- Secondary Storage (hard disk)
- Removable storage (flash drive, CD, etc)

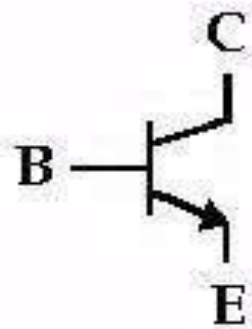
# Primary Storage: RAM (Random Access Memory)

- Made of Transistors



DRAM chips on the motherboard

# Transistors



- Collector, Base, Emitter
- Collector more positive than emitter (pnp transistors)
- B-E and B-C circuits act like diodes

$I_c = 50$  to  $250$  with  $100$  typically

$$I_e = I_c + I_b$$

$$V_b - V_e = .6 \text{ Volts}$$



- Used to amplify power
- Allow electrical signals to control other electrical signals (automatic computing)

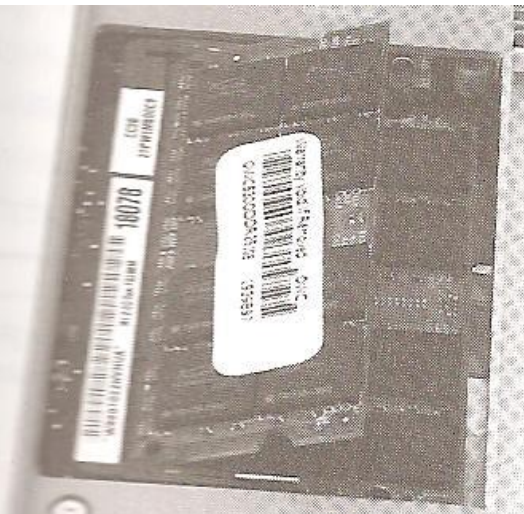
# Primary Storage: RAM (Random Access Memory)

- Programs are loaded into memory when run

- Made of Transistors
- Fast to access
- Expensive storage
- Only store data when powered

## Types of RAM

- DRAM
  - Dynamic Random Access Memory
  - Built as an integrated circuit
  - Can access any memory location
  - Multiple DRAMs are used together to hold a program
- SRAM
  - Static Random Access Memory
  - Faster than DRAM
  - Less densely packed
  - More expensive
  - Used in CPU cache memory



DRAM chips on the motherboard

# Secondary Storage: RAM (Hard Disk)

- Slower to access
- Holds data even when not powered

Platter

- Rotating part coated in magnetic material
- Read and write heads detect and change magnetic flux to store and read information



# Removable Storage



# Peripheral Devices (I/O Devices)

## Input Devices

- Keyboard
- Mouse
- Touch Screens
- Microphone

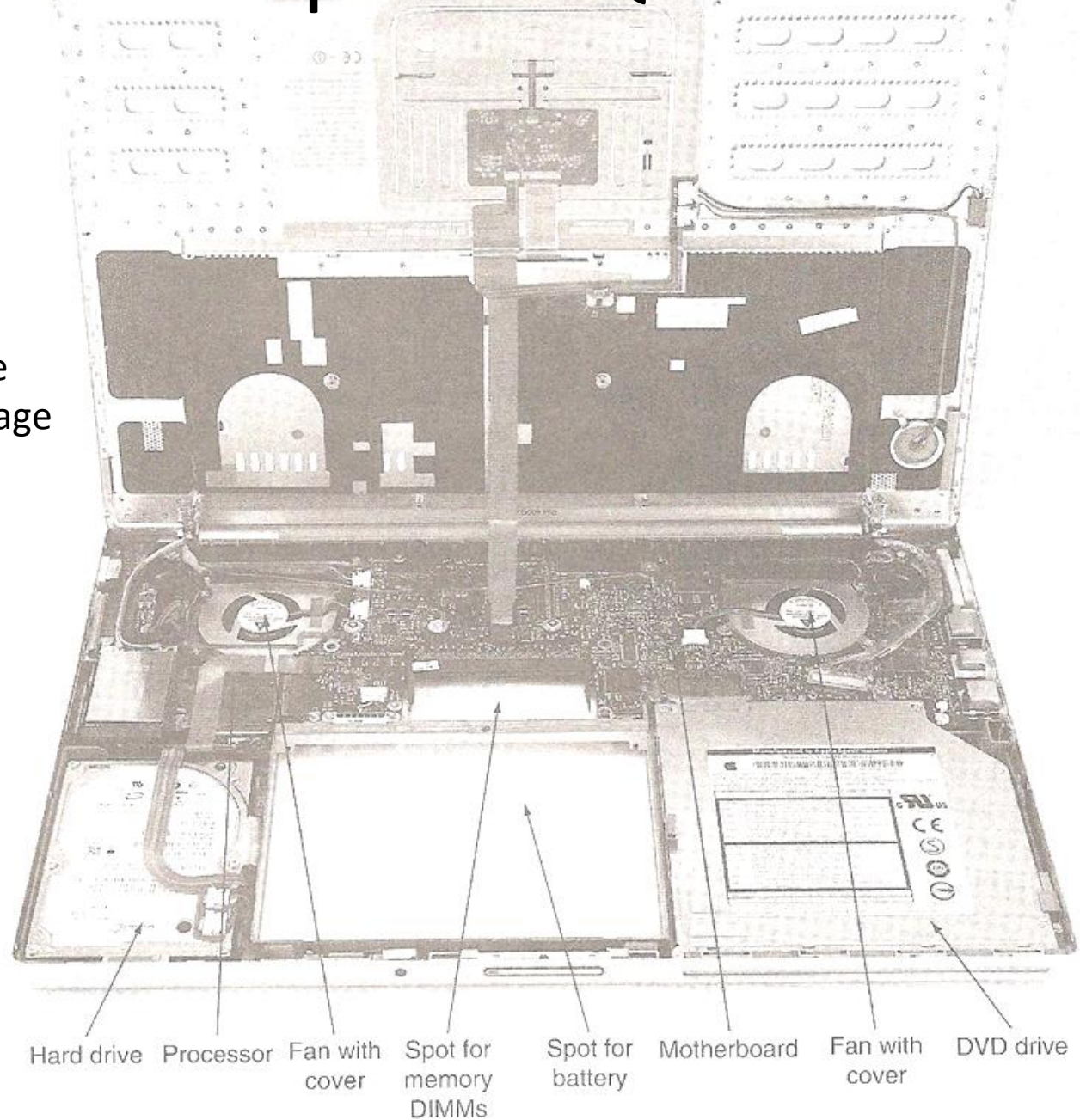
## Output Devices

- Printer
- Monitor
- Speakers



# Parts of a Computer: Questions?

- Physical Computer
  - Motherboard
  - CPU
  - Storage Devices
    - Primary Storage
    - Secondary Storage
- Peripheral Devices
  - Output Devices
  - Input Devices



# Software

## Computer Program

Definition: a sequence of instructions and decisions

- Stored on a hard disk, DVD, etc
- Moved into memory when run
- May read, modify, and write data
- May output to peripheral devices

## Computer Programming

Definition: the act of designing and implementing computer programs

# High Level Language

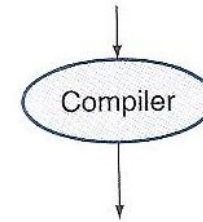
Definition: a portable language like C, C++, Java, or Visual Basic composed of words and algebraic notation

## Advantages of HLL

- You do not have to specify memory locations to store information
- You can compute  $5^{12}$  with one line of code, not 12 multiplications
- Avoid some steps

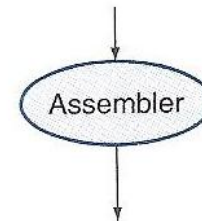
High-level  
language  
program  
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```



Assembly  
language  
program  
(for MIPS)

```
swap:
  muli $2, $5, 4
  add  $2, $4, $2
  lw   $15, 0($2)
  lw   $16, 4($2)
  sw   $16, 0($2)
  sw   $15, 4($2)
  jr   $31
```



Binary machine  
language  
program  
(for MIPS)

```
000000001010000100000000000011000
000000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
0000001111100000000000000001000
```

# Programming Ideas

Pseudo Code: informal description of steps to solve a problem

Algorithm: a sequence of steps that is unambiguous, executable, and terminating

Is this an algorithm?

- Begin with 15 coins. While you have more than 5 coins, add another coin to your pile.
  - Not terminating
- While there are books on the table, move the one whose author's name comes first in the alphabet to the next location on the shelf.
  - Unambiguous, executable, terminating.
- Go for a walk
  - Not unambiguous (where, length of time, when ...)
- For each person in line, give them a ticket.
  - Unambiguous, executable, terminating.
- Find the third largest number in this list: {1,5}
  - Not executable

# Example Algorithm

Problem: Alice and Bryan have graded a pile of papers and need to sort them alphabetically.

1. Alice and Bryan sort their own students' papers



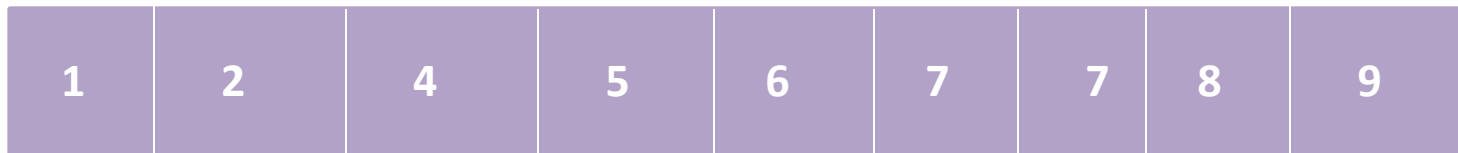
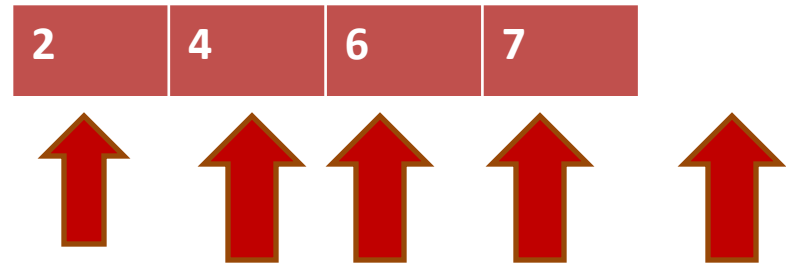
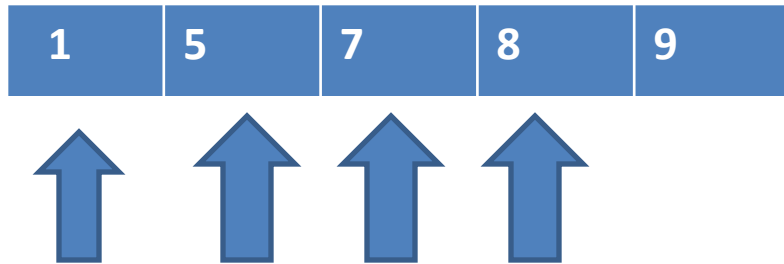
2. Maintain a current pointer in each list, start at the front of each list.

3. As long as neither pointer is at the end of the list repeat:

Of the two pointers, choose the paper which belongs first alphabetically, add it to the master pile, and advance that pointer.

4. When one pointer reaches the end of the list, copy all remaining elements from the other list to the master pile.

1. Begin with two sorted lists.
2. Maintain a current pointer in each list, start at the front of each list.
3. While no pointers are at the end of the list repeat: of the two pointers, choose the paper which belongs first alphabetically, add it to the master pile, and advance that pointer.
4. When one pointer reaches the end of the list, copy all remaining elements from the other list to the master pile.



Algorithm in Psudeo Code:

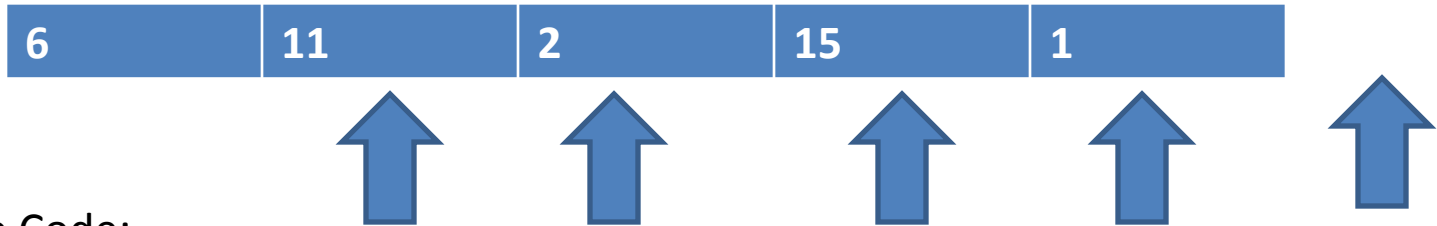
While both lists are non-empty

- Let  $A_i$  and  $B_i$  be the elements pointed to by their respective pointers
- Append the smaller of  $A_i$  and  $B_i$  to the output list
- Advance the current pointer in the list from which the smaller element was selected

Once one list is empty, append the remainder of the other list to the output.

# Second Example Algorithm

Problem: You want to find the largest number in a list of n numbers



Pseudo Code:

Let Max be the first number in the list

Look at each of the other numbers in sequence.

If the current number is larger than Max, change the value of Max

Max = ~~6~~ 15

# Software: Questions?

Computer Programming

Computer Program

High Level Languages

Pseudo Code

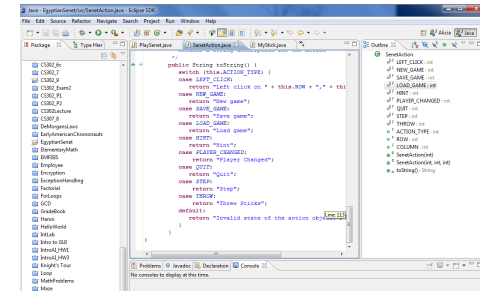
Algorithms

# Java Basics

- Introduction to Java
- Files
- Classes
  - Classes with a main method
  - Instantiable Classes
- Methods
- Errors
  - Compile Time Errors
  - Run Time Errors

# Introduction to Java

- Safe to use
- Portable
- Libraries exist
- Integrated Development Environment



- Case sensitive

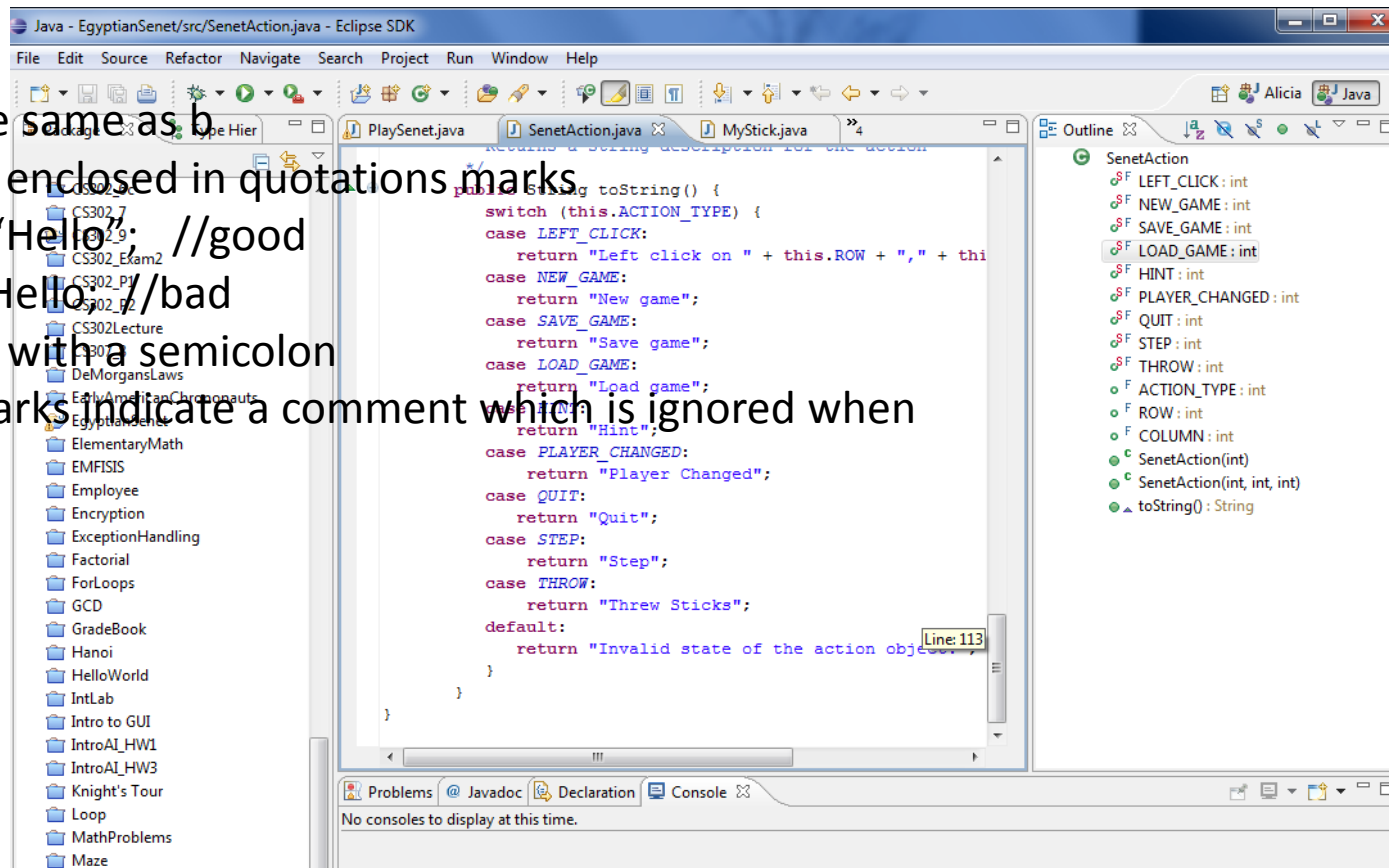
- B is not the same as b

- Strings must be enclosed in quotations marks

- String n = "Hello"; //good
  - String n = Hello; //bad

- End statements with a semicolon

- Double slash marks indicate a comment which is ignored when program is run



# Steps to write a Java Program

1. Write code and fix any problems noticed.
2. Compile the program.
3. Run the program.
4. Back up your work.

In eclipse, the program is automatically compiled and run when you press the “run” button.



When running programs from the command line, you must compile and then run the program.

```
C:\> Command Prompt
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Alicia>javac Robot.java
```

```
C:\Users\Alicia>iava Robot
```

# Java Files

## Source Code

- Typed code, readable by humans
- Stored as filename.java

## Compiled Code

- Stored as filename.class
- Created by the compiler from the source code
- Readable by the JVM

# Java Classes (Background)

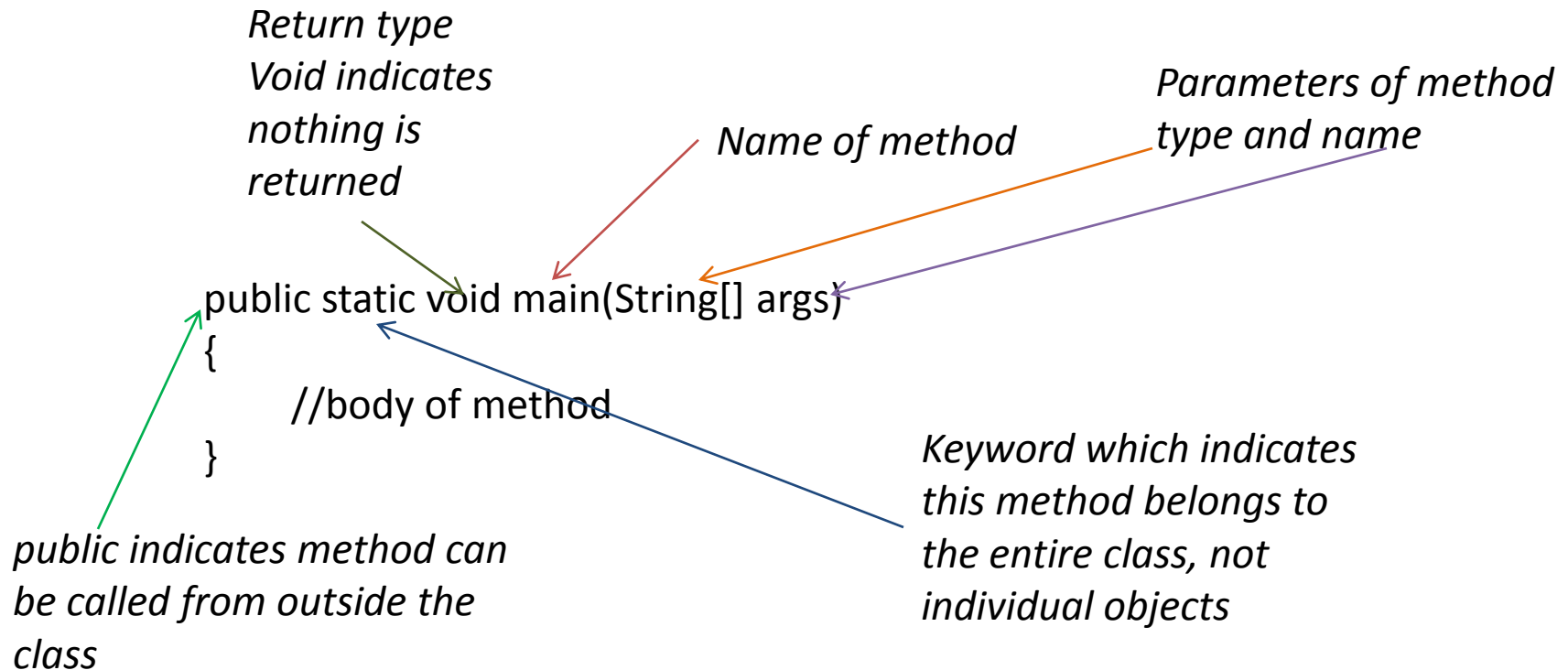
- Class- the fundamental building block of a Java program
- Class declaration:

```
public class NameOfMyClass  
{  
    //body of my class  
}
```

- Class names are capital by convention
- Two Types of classes
  - Instantiable Classes
    - The “blueprint” for an object you intend to create
  - Non-instantiable classes
    - A container for your program

# Java Methods (Background)

- The code that controls the program's actions
- Names are lower case by convention
- Main method
  - Required in all non-instantiable classes
  - Automatically run when the program is started
  - Declaration:

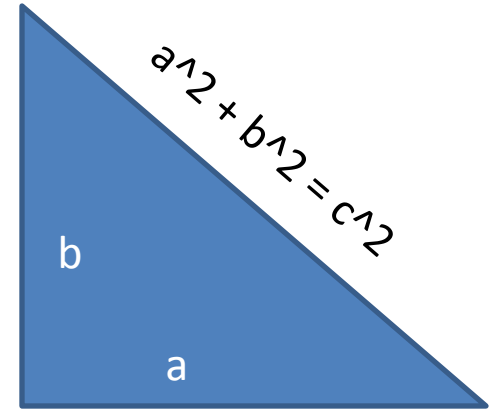


# Non-Instantiable Class Example

Problem: calculate the hypotenuse of a right triangle

```
public class RightTriangle
{
    public static void main(String[] args)
    {
        double hypot = getHypotenuse(1.1,2.1);
        System.out.println("hypotenuse is : " + hypot);
    } //end main method

    public double getHypotenuse(double side1, double side2)
    {
        double side1Sq = side1*side1;
        double side2Sq = Math.pow(side2,2);
        double cSq = side1Sq+side2Sq;
        return Math.sqrt(cSq);
    } //end method getHypotenuse()
} //end class RightTriangle
```



Result is 2.37065...

# Instantiable Class Example

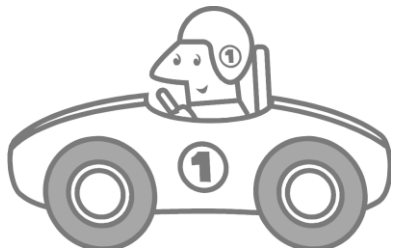
Problem: write a race-car driving game



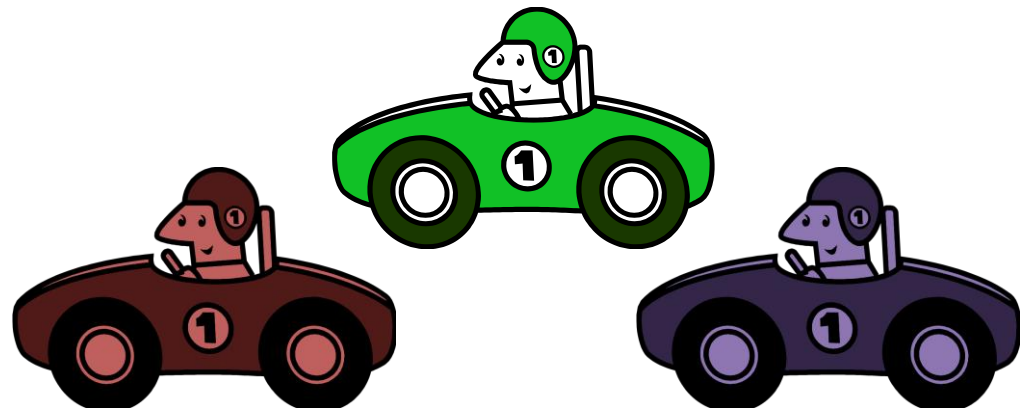
What do we need for our game?

- Cars
- A race track
- Possibly coins that can be collected for points
- Possibly hazards that must be avoided

Each of these is like a blueprint for how to make a specific object (a car, for example). We can create as many “instances” of the class as we want.



Blueprint of car



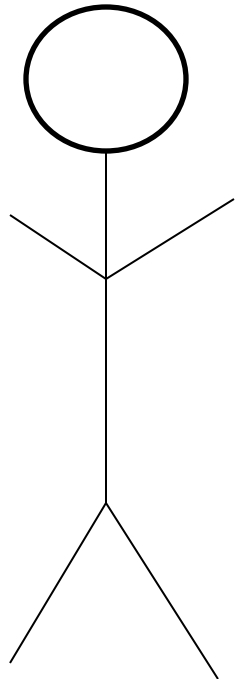
Instances of the car class

# Instantiable Class Example 2

Problem: CEO of company wishes to pay employees using a computer program.

```
public class Employee
{
    private String name;
    private double wage;
    private double hoursWorked;

    //possibly more data and methods
}
```



```
public class PayEmployees
{
    public static void main(String[] args)
    {
        Employee Mary = new Employee("Mary");
        Employee Joe = new Employee("Joe");

        //.....
    }
}
```



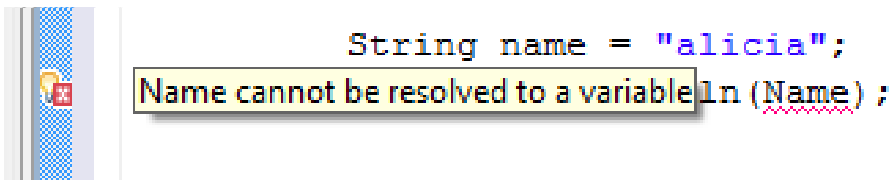
# Compile Time Errors

- A violation of the programming language rules that is detected by the compiler.
- “syntax error”

• Example 1: `System.out.println("hello");`  
~~`System.out.println("hello");`~~

• Example 2:  
`double hypot = 12.4;`  
`System.out.println(hypo);`

- Program will not compile and therefore will not run
- Indicated in Eclipse:



# Run Time Errors

- a mistake that causes the program to take an action that the programmer did not intend.
- Program is syntactically correct
- Error will not be found by compiler
- Program will compile
- “logic error”

- Example 1:

```
public double getHypotenuse(double side1, double side2)
```

```
{  
    double side1Sq = side1*side1;  
    double side2Sq = Math.pow(side2,2);  
    double cSq = side1Sq+side2Sq;  
    return cSq;  
}
```

```
//end method getHypotenuse()
```

Error:  $\text{hypot} \neq a^2 + b^2$   
 $\text{hypot} \neq \sqrt{a^2 + b^2}$

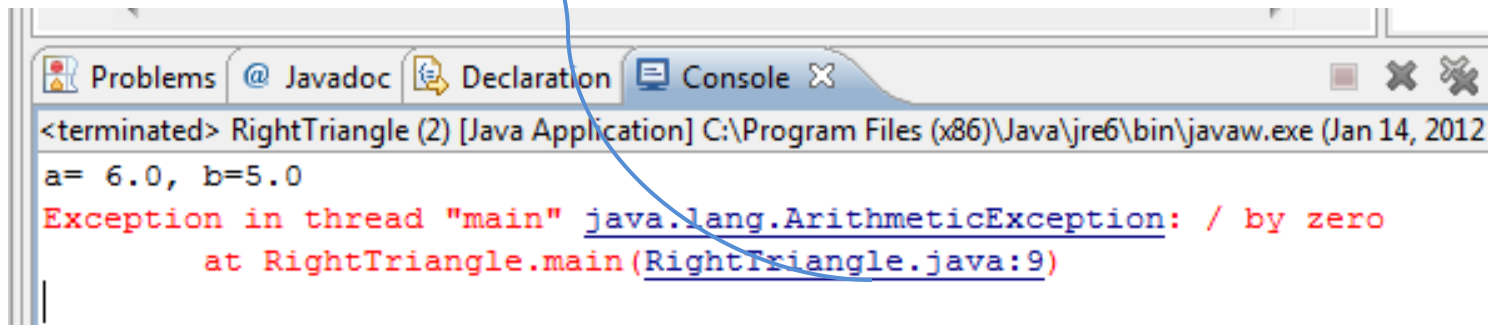
- Exceptions may be generated
  - Program will stop at the point the exception was generated
  - Error message will be displayed

# Run Time Errors

- Exceptions may be generated
  - Program will stop at the point the exception was generated
  - Error message will be displayed

• Example:

```
public static void main(String[] args)
{
    double a=6; double b=5;
    System.out.println("a= "+ a+ ", b=" +b);
    double c = a+b;
    System.out.println(1/0); ←
    System.out.println("a+b=" + c);
}
```



# Java Basics: Questions?

- Introduction to Java
- Files
- Classes
  - Classes with a main method
  - Instantiable Classes
- Methods
- Errors
  - Compile Time Errors
  - Run Time Errors



DEPARTMENT OF  
**Computer Sciences**  
UNIVERSITY OF WISCONSIN-MADISON