Procedures

Methods

Lecture Objectives:

1) Define procedure
2) Explain the MIPS calling convention for procedure calls
3) Define the term program counter (PC)
4) Explain the relationship between the caller and callee in a procedure call
5) Explain the purpose for the stack when invoking procedures
6) Explain the relationship between characters and strings.
7) Explain the three main mechanisms for representing strings in memory. — May not cover today
8) Explain the relationship between the 5 MIPS addressing modes.

Compare and contrast:
What is different here?

```
Public static void main(String[] args)
{
  for (int x = 0; x < 10; x++)
  {
    System.out.println(x);
  }
}
```

```
Public static void main(String[] args)
{
  for (int x = 0; x < 10; x++)
  {
    printit(x);
  }
}

Public static void printit(int number)
{
  System.out.println(number);
}
```

Call a procedure

Go here and return

Does not have an extra method

New method.

Procedure: Going to another segment of code.

Definitions

- Procedure
  - A stored subroutine that performs a specific task based on the parameters which it is provided
- Return address
  - A link to the calling site that allows a procedure to return to the proper location once it is completed.
- Program counter
  - The register containing the address of the instruction in the program being executed
- Stack
  - A data structure for spilling registers organized as a last in first out queue
- Stack pointer
  - A value denoting the most recently allocated address in a stack that shows where registers should be spilled or where registers can be found
- Push
  - Add an element to the stack
- Pop
  - Remove an element from the stack.

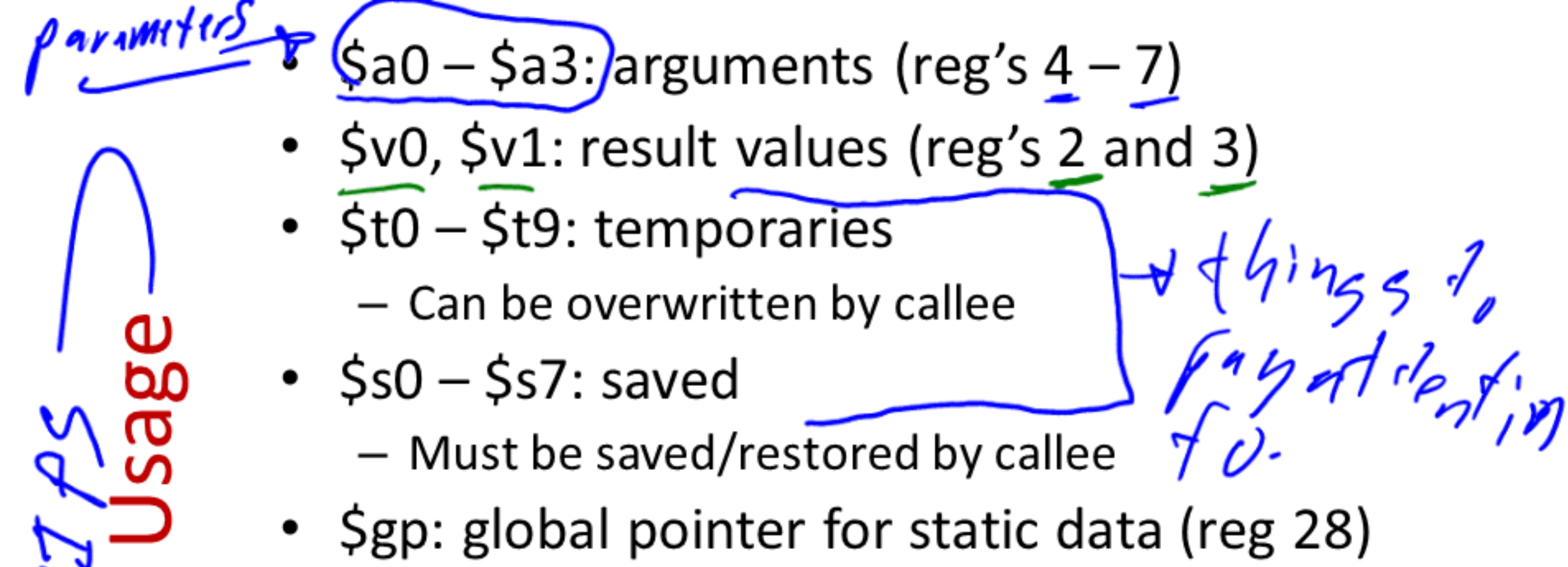
- Steps required

1. Place parameters in registers
2. Transfer control to procedure
3. Acquire storage for procedure
4. Perform procedure's operations
5. Place result in register for caller
6. Return to place of call

Procedure Calling

Register Usage

- $a0 – $a3: arguments (reg's 4 – 7)
- $v0, $v1: result values (reg's 2 and 3)
- $t0 – $t9: temporaries
  - Can be overwritten by callee
- $s0 – $s7: saved
  - Must be saved/restored by callee
- $gp: global pointer for static data (reg 28)
- $sp: stack pointer (reg 29)
- $fp: frame pointer (reg 30)
- $ra: return address (reg 31)



Calling a method

Two instructions

to keep track of.

Procedure Call Instructions

- Procedure call: jump and link

  `jal ProcedureLabel`

  – Address of following instruction put in $ra

  – Jumps to target address

- Procedure return: jump register

  `jr $ra`

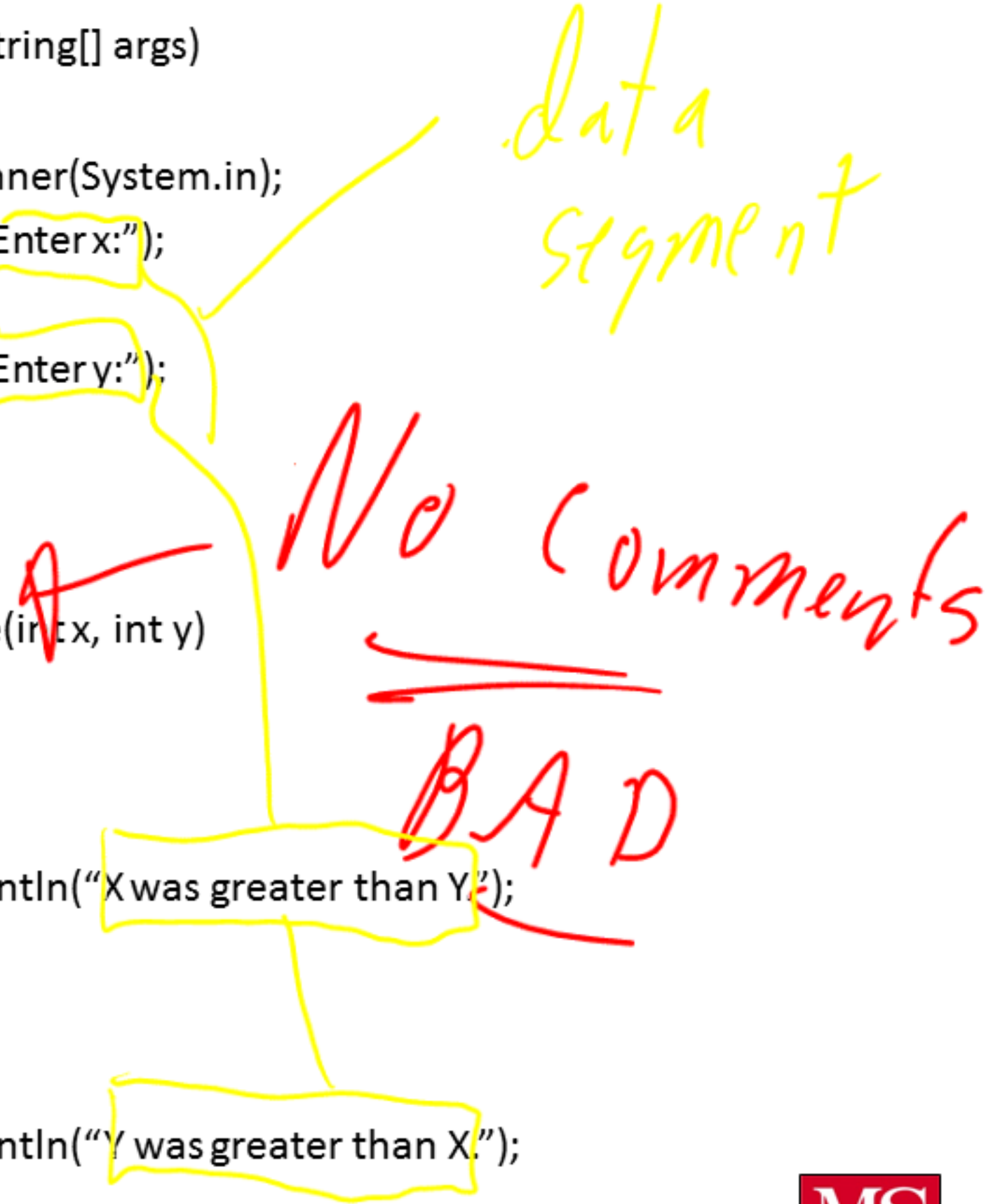
  – Copies $ra to program counter

  – Can also be used for computed jumps

    - e.g., for case/switch statements

Lets look at an example (Java Code First)

```
public static void main(String[] args)
{
    Scanner s = new Scanner(System.in);
    System.out.println("Enter x:");
    Int x = s.nextInt();
    System.out.println("Enter y:");
    Int y = s.nextInt();
    analyze(x, y);
}
public static void analyze(int x, int y)
{
    if (x > y)
    {
        System.out.println("X was greater than Y.");
    }
    else
    {
        System.out.println("Y was greater than X.");
    }
}
```



How are strings represented in assembly?