Memory Work

For Unit 2

```
//Counting Loop
for (int i=0; i<15; i++) {
    System.out.print(i+" ");
}
System.out.println();
```

```
//Sandwich loop
char end ='n';
while(end=='n') {
   System.out.println("Hello");
   end = IO.inputChar("End? (y/n) ");
}
```

Parts of the Loop

1. ILSV - Initialize Loop Stopping Variable
2. TLSC - Test Loop Stopping Condition
3. STR – Steps to Repeat
4. PTTLSC – Progress to the Loop Stopping Condition

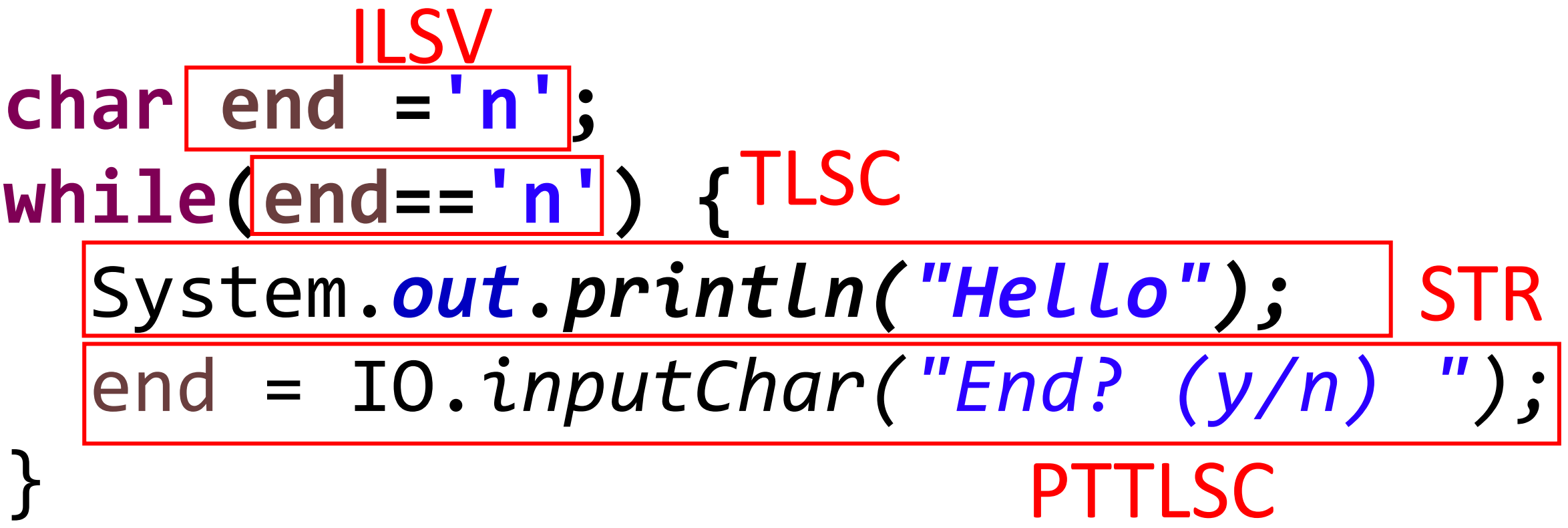
Parts of the Loop

ILSV TLSC PTLSC

```
for (int i=0; i<15; i++) {
    System.out.print(i+" ");
}
System.out.println();
```

STR

Parts of the Loop



```
//void method
public void methodName () {
    code;
}
```

Parts of the Method

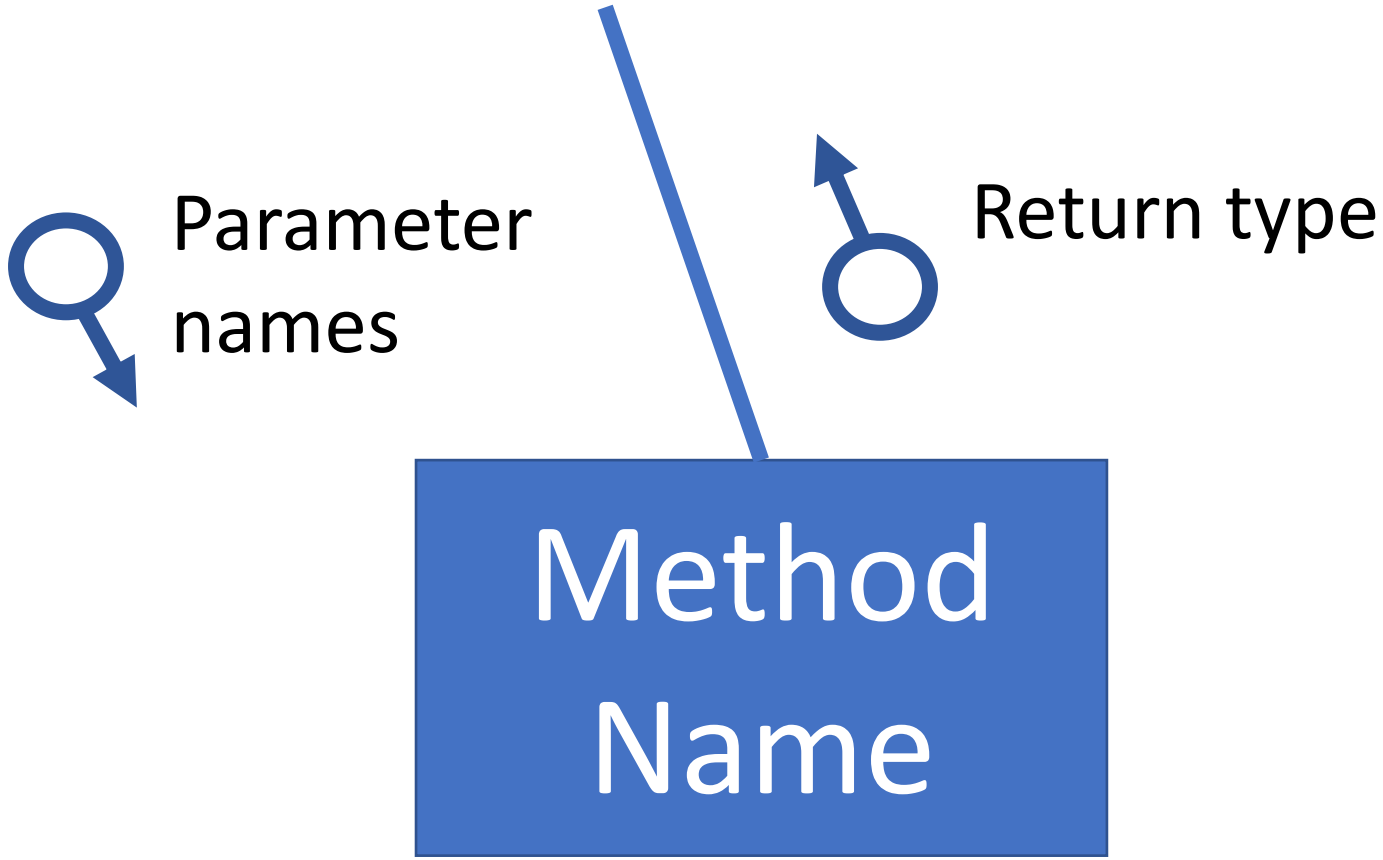
1. Method signature
2. Return type
3. Method name
4. Parameter
5. Parameter name
6. Parameter type

```
//Complex method
public returnType methodName (paramType paramName) {
    code;
    return something;
}
```

Product Development Life Cycle

1. Analysis – Proposal for Solution to Problem
2. Design – Detailed Plan, Planning Diagrams
3. Code – Complete program, fully tested, commented.
4. Reflection – Direction for next version, sales, advertising, support for current version.

Structure Chart Pieces



Unit 2 Summary

Test Preparation

Loops

- Repetition statements (unlike ifs; ifs decide things, loops repeat them)
- Control structures (like ifs, both have Boolean expressions that change the flow of the code)
- Inside of cutting and pasting our code over and over, we use a loop. It will run the code multiple times.
- There is no such thing as an ifloop. There are ifs and there are loops.

Java's Loops

Java has three kinds:

- For Loops (for a set number of times)
- While Loops (until something happens)
- Do While Loops (discover on your own for the project)

Errors in the Parts of a Loop

- All of the parts of a loop have to be present and they have to work together.
- If they aren't then,
  - The loop might not run at all.
  - The loop might become an infinite loop and run forever

While Loops vs. For Loops

- For Loops are great for “counting” situations – if you know that you need to do something a specific number of times, use a for loop.
- However, if the user gets to pick how many times it happens, use a while loop.
- While loops can easily stop when:
  - The user chooses to quit
  - The game is over
  - There is a tie
  - The user answers correctly
  - The input is valid (no longer an error)

More about the pieces of a loop!

1. Initialize the loop variable

```
char end = 'n';

while (end == 'n')
{
   System.out.println ("Hello");

   end = IO.inputChar ("End? (n/y) ");
}
```

- Happens ONCE
- Happens before the loop is run
- Declares the loop stopping variable.
- Gives the loop stopping variable a value that makes the loop stopping condition true… so that the loop will run.

More about the pieces of a loop!

```
char end = 'n';

while (end == 'n')
{
   System.out.println ("Hello");

   end = IO.inputChar ("End? (n/y) ");
}
```

2. Test the stopping condition

- Uses the loop stopping variable (end in this case)
- Is also a Boolean Expression
- If the expression is TRUE, the loop runs
- If the expression is FALSE, the loop stops

More about the pieces of a loop!

```
char end = 'n';

while (end == 'n')
{
   System.out.println ("Hello");

   end = IO.inputChar ("End? (n/y) ");
}
```

3. Steps to repeat

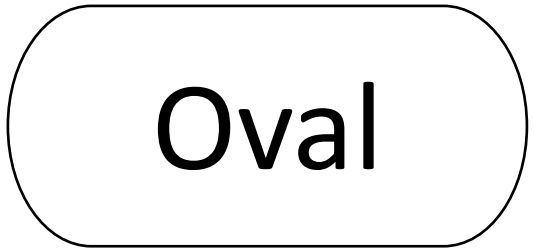
- Can be many statements… even an entire game
- Runs over and over again when the loop stopping condition is true
- Inside the { }

More about the pieces of a loop!

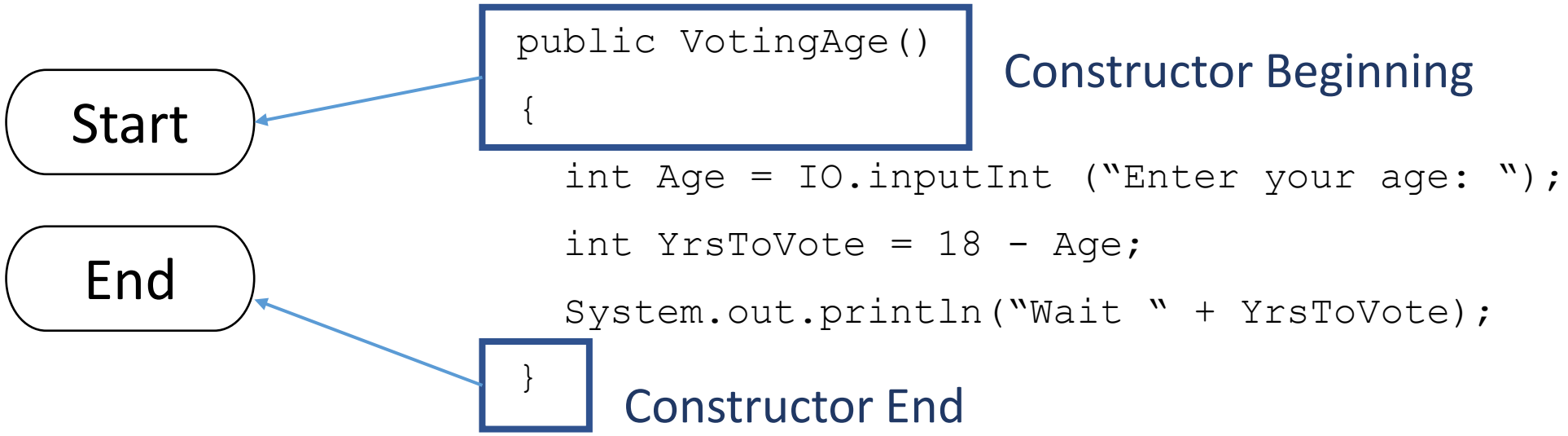
```
char end = 'n';

while (end == 'n')
{
   System.out.println ("Hello");

   end = IO.inputChar ("End? (n/y) ");
}
```

4. Move towards the stopping condition

- If this part is missing, the loop will run forever
- This part is where the user has a chance to get out
- The loop stopping variable (end in this case) is changed on this line
- It must be inside the { }



Terminal



- One Start and One End for each program. No more, no less.
- Only holds “Start” and “End”
- Only one arrow comes out of the Start. No arrows for End.

Input

IO lines

```
public VotingAge()
{
  int Age = IO.inputInt ("Enter your age: ");
  int YrsToVote = 18 - Age;
  System.out.println("Wait " + YrsToVote);
}
```

- Used for IO lines.
- Only write “Get” + variable name
- Don’t write the prompt (question)
- Only one arrow comes out of it.

Rectangle

Processing

YrsToVote = 18-Age

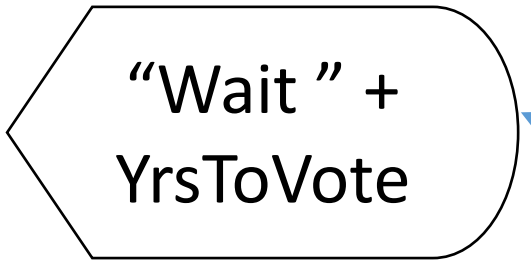
```
public VotingAge()
{
   int Age = IO.inputInt ("Enter your age: ");
   int YrsToVote = 18 - Age;
   System.out.println("Wait " + YrsToVote);
}
```

Math lines

- Used for Math Lines.
- Leave out the variable type, but write everything else
- Only one arrow comes out of it.

Pencil

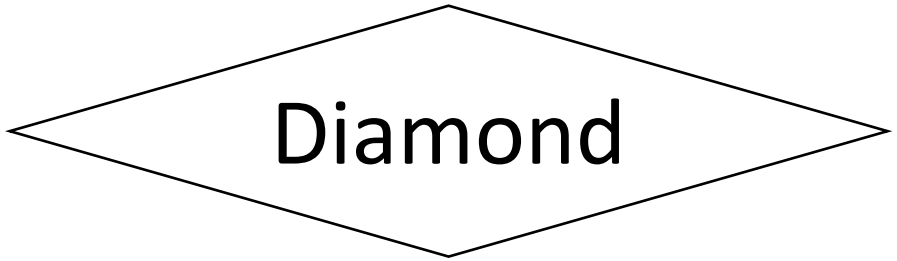
Output



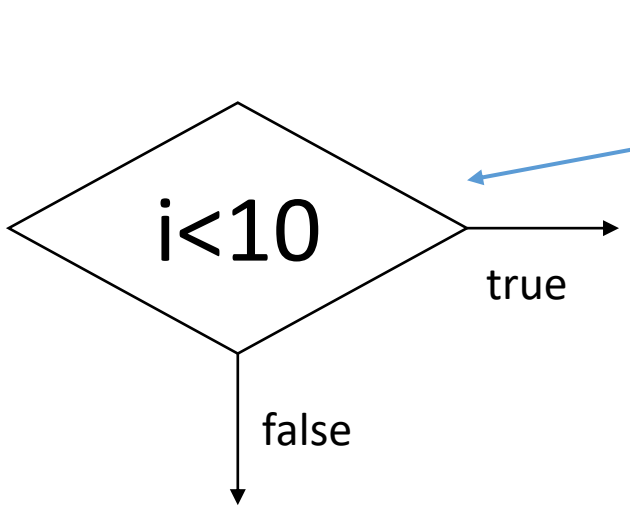

```
public VotingAge()
{
   int Age = IO.inputInt (“Enter your age: “);
   int YrsToVote = 18 - Age;
   System.out.println(“Wait “ + YrsToVote);
}
```

System.out lines

- Used for System.out.println lines.
- Leave out the System.out.println(); but write everything else
- Only one arrow comes out of it.



Boolean Expression




```
for (int i=0; i<10; i++)
{
   int Age = IO.inputInt ("Enter your age: ");
   int YrsToVote = 18 - Age;
   System.out.println("Wait " + YrsToVote);
}
```

- Used for Boolean Exressions – ifs && loops.
- Exactly 2 arrows come out of them. One is labelled true and one is labelled false.

PDLC

- Stands for: Product Development Life Cycle
- It is the stages that large computer companies use to make a big computer program.
- For example, a game like Counter Strike will have hundreds of programmers working for about 2 years on the code.
- How do so many people work together effectively? Structure and pre-planning.

Phases of PDLC

- There are 4 phases of the PDLC
- A. Analysis: the overall idea of the game is developed.
- B. Design: the detailed plan for the game is created.
- C. Coding: the program is coded and tested.
- D. Reflection: the program is sold and the sales performance is evaluated.

Dumb Ways to Die

- *An example of an innovative **Analysis** Phase*
- In Melbourne, there were many near-miss accidents with trains. Public service announcements had no effect.
- In their analysis phase, Metro Trains decided to create a silly app with train accidents hidden inside it.
- Results? Huge success.
  - 30% reduction in near-miss accidents.
  - $50 million in advertising for a few thousand dollars.

Candy Crush

- *An example of an innovative **Design** Phase*
- The artwork and characters were targeted at an untapped market: women and professionals.
- King software also decided to use the freemium model (free to play, pay to level up quickly) to generate revenue.
- These two design decisions were very successful: in 2013, Candy Crush made about $1 M per day.

Avatar (2009)

- *An example of an innovative **Coding** Phase*
- The coding to create Avatar's rendering was so complex it took 900 coders several years to create it.
- This huge team required very detailed plans to allow them to each code a small piece and then fit all pieces together.
- This team was very successful.
  - Highest grossing movie to date.
  - Made $1B in 19 days.
  - Also first movie to create photo-real CGI people.

GTA 5 (2013)

- *An example of an innovative **Reflection** Phase*
- Total development cost of $265 million. $128 million of that was marketing.
- Used old media (billboards, news) and new media (social, streaming) to market the game.
- Results? Very successful.
  - Fastest selling entertainment product to that date.
  - $1B in 3 days

Analysis Phase Summary 1

- *End result:* Overall idea for the game is decided
- Includes:
    1. Define the problem
    2. Brainstorming
    3. Interviews
    4. High concept statements
    5. Pitch the concept
    6. User focus groups
    7. Develop a specifications list
- *Jobs?* Analyst, CEO

Design Phase

2

- *End result:* Detailed plans for the game
- Includes:
    1. Flow charts, Structure Chart
    2. Memory Diagrams
    3. Detailed character designs
    4. Music composition & recording
    5. Artwork
    6. Writing story line
    7. Design prototypes
    8. Level design
- *Jobs?* Writer, Musician, Graphic Designer, Artist, Code Designer, Voice Artist

Coding Phase 3

- *End result:* The game is coded and tested.
- Includes:
    1. First Playable (basic code)
    2. Add comments
    3. Write ifs, loops, methods
    4. Alpha test (internal testers)
    5. Code Freeze (stop adding new)
    6. Beta test (external testers)
    7. Gold Master (complete game)
- *Jobs?* Programmer, Lead Programmer, Tester

Reflection Phase 4

- *End result:* Game has been marketed and sold.
- Includes:
    1. Marketing
    2. Sales
    3. Reflection
    4. Plan sequel
- *Jobs?* Sales Analyst, Advertiser

Method

- A subprogram.
- Used to break a larger program into smaller pieces.

Method Signature

- First line of the method
- Very important because it specifies all of the input and output of the method AND it’s name

Method Name

- Like a variable name, but for the method
- Method must be named using naming rules
  - No keywords
  - Can’t START with a number
  - No spaces
  - No special characters

Return Type

- The value sent OUT of the method.
- The value is sent back using the “return” line
- It must be the same type specified in the method signature.

Parameter

- A variable sent into a method
- It has a type(possibly a view type) and a name (follows Id's naming rules
- INPUT of the method

Method Calls

- Outside the method, possibily in the constructor, you use the method name to “activate” the method
- During the method call, you pick the parameters and you catch the returned values