**Introduction to MCP**

CPS 1  
Summer 2002  
Shannon Pollard

---

**Algorithms**

• Algorithm: A detailed set of instructions for solving a problem that:
  – Has a finite number of steps
  – Takes a finite amount of time

---

**Multiple-Choice Programming**

• Way to write out algorithms
• Show the level of detail you need to use
• Some things provided for you
• Not learning Java
• Focus on universal aspects of programming

---

**Lessons Learned from PB&J: Variables**

• Need a way to name things (bread slice 1)
• Variables: Names assigned to things so they can be referred to later.
  int x = 4;
• Remember Algebra? Kind of, but not quite…
  Thing on left gets CHANGED!!
  int y,z;
  y = 4; (“Y gets 4”)
  z = y;
  y = 6;
  z = z+1;
Variables, Cont’d

- Variables have Type- what kind of thing is it?
  - int, double
  - boolean – things that are either true or false
  - Strings – “Put anything in quotes!”
  - Objects – Anything!!!
- You first must tell the type of a variable, then you can give it a value
- Examples:
  ```java
  int x = 4;
  String answer = “Here it is!”;
  boolean b = (x==3);
  ```

Lessons Learned from PB&J: Subroutines

- Allow a set of steps to be done over again with little change, without rewriting everything (Now repeat steps 3-15 with bread slice 2 and jelly)
- Subroutines may or may not take input (parameters) or give an answer (return something).
- Examples:
  ```java
  int y = addTwo(42);
  String s = whatsMyLine();
  quit();
  ```

Subroutine Specifics

- **Header** – information at beginning of subroutine, including its name
- Header has type of thing returned, or **void** keyword
- **Parameters** – tell type of inputs and gives them local names
- Must have **return** statement unless return type void
- Can be used by name by other functions

Lessons Learned from PB&J: Loops

- Loops are used to do a sequence of steps some number of times, or only in certain situations (Repeat steps 3-10 5 times; Repeat step 4 until the jelly is spread, do step 5 only if there is more bread.)
- Need to use booleans – things that are either true or false
  ```java
  (x<=5)
  myString.equals(“Shannon”)
  (obj!=null)
  ```
Loops, Contd

- If Statements
  
  ```
  if (BOOLEAN) {
    STATEMENTS
  } else {
    STATEMENTS
  }
  ```

If Example

```cpp
int x = 4;
x = x + 14;
if (x < 18) {
  x = 0;
} else {
  x = 1;
}
return x;
```

Loops, Cont’d

- While Loops
  
  ```
  while (BOOLEAN) {
    STATEMENTS
  }
  ```

While Loops

```cpp
int x = 4;  // Initialization
while (x < 10) {  // Check
  x = x + 1;  // Update
}
return x;
```
public int addThree(int number) {
    return number + 3;
}

public void writeOut(String line) {
    System.out.println(line);
    String myName;
    myName = "Shannon";
    writeOut(myName);
}

public List listCPS1(List master) {
    List answer = new List();
    List cps1 = getClassList(cps1);
    int last = master.length();
    int index = 1;
    while (index <= last) {
        Student s = master.getMember(index);
        if (cps1.contains(s)) {
            answer.add(s);
        }
        index = index + 1;
    }
    return answer;
}

*MCP Format*

- Most lines specified that you need to write
- You must fill in the correct variables or statements
- You make your own variables as needed
- Things in CAPS are supposed to be substituted!!
- Other things must be typed exactly
- Order matters!
- Cheat Sheet provided