

CS 61b – Summer 2005

Homework #2 – Due July 5th at the beginning of class

This homework is to be done individually. You may, of course, ask your fellow classmates for help if you have trouble editing files, compiling programs, or using the submit program. Do not share code or specific parts of answers. Please see the syllabus for guidelines on appropriate interaction with your fellow classmates.

Homeworks will often require written answers as well as small programs (Java source files). *All written answers are to be placed in a file called **writeup.txt**.* This file should be a plain text file. Make sure this file is readable by “wordpad” or “emacs” or similar text-only editor. No credit will be given for written answers stored in a non-text format. Please ask your TA for help if you have trouble.

When you are finished with the homework, you will submit it electronically. First put all of the files you want to submit into their own directory (including writeup.txt). Change to that directory, and then type “submit hw2”

**For this homework you can put all your work into writeup.txt, or you can include additional files as well. If you include additional files, make sure to reference those files in the appropriate place in your writeup.txt file so we can find them and will know which files have which answers. Thanks.**

If you can't complete the entire homework, and it complains about you not having certain files, then just edit a blank file with that name, and then save it. This will submit an empty file, but the submit program should then work.

## **Problem 1**

1.1) The process of turning Java source files (ending in .java) into machine-readable files that can be executed is called \_\_\_\_\_

1.2) Given these lines of code:

```
int i = 3;  
float f = 3.14f;  
long L = 23L;  
double d = 22.3;
```

Which of these statements will compile?

```
i = L;
L = i;
f = i;
d = f;
f = d;
f = (float) d;
f = (int) d;
d = (double) ((float) d);
```

## Problem 2

2.1 What is the result of the following lines of code?

```
for (int i = 0; i < 10; i = i+2) {
    if (i % 3 == 1)
        System.out.println(i * 2);
}
```

2.2 Write the following switch statement as nested and daisy-chained if statements:

```
String val;
int i = getSomeInteger(), k = getSomeOtherInteger();
```

```
switch (i) {
    case 0:
        val = "str1";
        break;
    case 1:
        val = null;
        break;
    case 2:
        if (k % 3 == 0) {
            val = "str2";
        }
        break;
    case 3:
    case 4:
    case 5:
        val = "str3";
    default:
        val = "str4";
}
```

## 2.3 What is the difference between

```
for (init; test; update) {  
    statements;  
}
```

and

```
init  
while (test) {  
    statements;  
    update;  
}
```

(Hint: what could *statements* contain that would affect the flow in a loop? We talked in class about two keywords which, when put into loops, affect the way they execute).

## Problem 3

3.1 Fill in the following method body. This method should return an array of size N, where the i'th entry consists of the sum of the first i integers (starting at 0).

So for example, if N == 4 the array returned would be:  
{ 0, 1, 3, 6 }

```
public int[] sumUpTo(int N) {
```

```
}
```

3.2 Declare an array of String object references called "stars" Make the i'th element of that array refer to a String of length i that contains only the \* character.

For example, the 3<sup>rd</sup> element of the stars array would point to a String object representing "\*\*\*"

## Problem 4

In this problem, we are going to design and build a “StampMachine” class. A stamp machine is a vending machine which takes money as input, and gives the user stamps as output. The StampMachine class has the following methods:

- A default constructor
- an “addStamp” method which takes a Stamp object as input and has a **void** return type
- a “buyStamp” method which takes an “int” argument representing how many cents the stamp is worth. This method returns a Stamp object of that value (if such a stamp is in the machine). If there is no stamp of that value in the machine, then the method return **null**. Once a stamp is purchased it should be removed from the machine
- a “getMoney” method which takes no arguments. It returns the amount of money stored in the machine (from when people bought stamps) in the form of an “int”. A side-effect of this method is that the current count of how much money is in the machine is reset to 0.

Example:

```
StampMachine sm = new StampMachine();
Stamp s1 = new Stamp(37);
Stamp s2 = new Stamp(37);
Stamp p1 = new Stamp(385);
sm.add(s1);
sm.add(s2);
sm.add(p1);
Stamp myStamp = sm.buyStamp(385);
Stamp myStamp2 = sm.buyStamp(37);
Stamp myStamp3 = sm.buyStamp(33);    // returns null

int totalMoney = sm.getMoney();      // totalMoney is now 422
sm.getMoney();                       // returns 0 now
```

Here is the Stamp class (NOTE: It is very much like the Book class we discussed in lecture. Note the equals() method, meaning that the Java API library will usually compare two Stamp objects based on their value, in this case.

```
public class Stamp {
    private int value;
```

```

public Stamp(int v) {
    value = v;
}

public String toString() {
    return new String(value);
}

public boolean equals(Object o) {
    if (!(o instanceof Stamp)) {
        return false;
    } else if (this == (Stamp) o) {
        return true;
    }
    Stamp s = (Stamp) o;
    return this.value == o.value;
}
}

```

4.1 How do you want to internally keep track of 1) the stamps in the system, and 2) the amount of money in the machine? In class we talked about using Java arrays (so Stamp[]) and we talked about ArrayList. Choose either of these options, or if you prefer to do it a different way, that is ok too.

4.2 Write out the class definition (leave the method bodies empty for now)

4.3 Now write definitions for each of the methods in your class.

## Problem 5

Consider the following classes:

```

public class Animal {
    public void sleep();
    public void wakeUp();
}

```

```
public class Feline extends Animal {
    public void roam();
}
```

```
public class Cat extends Animal {
    public void makeNoise();
}
```

```
public class Lion extends Animal {
    public void makeNoise();
}
```

5.1 Which of the following lines will compile?

Animal a = new Animal();                      Compiles / Doesn't compile

Feline f = new Feline();                      Compiles / Doesn't compile

Animal x = new Feline();                      Compiles / Doesn't compile

Feline y = new Cat();                      Compiles / Doesn't compile  
y.makeNoise();

5.2 When I call:

```
Cat c = new Cat();
```

5.3 Consider:

```
public class A {
    public void foo();
}
```

```
public class B extends A {
    public void foo();
}
```

5.3 What is it called when class B provides a different foo () method than its superclass? \_\_\_\_\_

5.4 When I call

```
B b = new B();
b.foo();
```

which foo() gets called?

Is the same true for when I access field variables?