Name: Andrew Januszk

Stage 1

Fill in The Blank

1. A(n) assembler translates assembly code to object code while a(n) compiler translates high level language to object code.

2. When you want to watch to see what you code it doing, it is useful to set a(n) breakpoint.

3. The keyword we use for a conditional statement is if.

4. Putting two strings together is concatenation.

5. A(n) translater translates source code to machine code at run time while a(n) compiler does that once before you can run the code.

6. A declaration of a variable requires two words: the type and variable name.

7. The words that the compiler understands are called keywords.

8. In an assignment statement, the two sides of the equal sign aren't the same thing. The left hand side is the variable name and the right hand side is an expression

x is assigned the value 42

9. We read this assignment statement: x = 42; as _____.

Content Questions

1. What are the critical points in the definition of “algorithm”?

The critical points in the definition of "algorithm" are that they are steps that always lead to the correct solution

2. What do we mean by the “target machine” of an algorithm?

By saying the "target machine" of an algorithm we mean the steps taken in the algorithm will lead to the correct solution on the target machine

3. English is made of 24 letters and some punctuation symbols. What is the language that our computers understand made of?

Our computers understand machine code.

5. What is the difference between an assembly language and a high level language?

An assembly language translates one statement to exactly on machine instruction while high-level language translates into many machine instructions.

7. What is the difference between source code and object code?

Source code is a program written in either a high-level language or assembly language while object code is the output of a compiler or an assember.

8. In your own words, explain this sentence, “The machine is short sighted and sequential.”

The machine does not remember what was once assigned to a variable and it only reads code one line at a time.

10. In your own words, what is the Java Virtual Machine and why do we have it?

The Java Virtual Machine is an interpreter that reads the code when it needs it and outputs it as object code and we have it because it allows the computer to read what is being output.

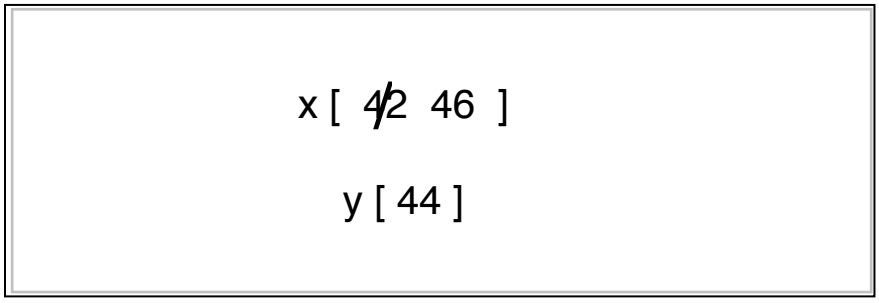
Memory Diagrams

For the following snippets of code, draw the resulting memory diagram. Here is one way to make and insert the pictures you need:

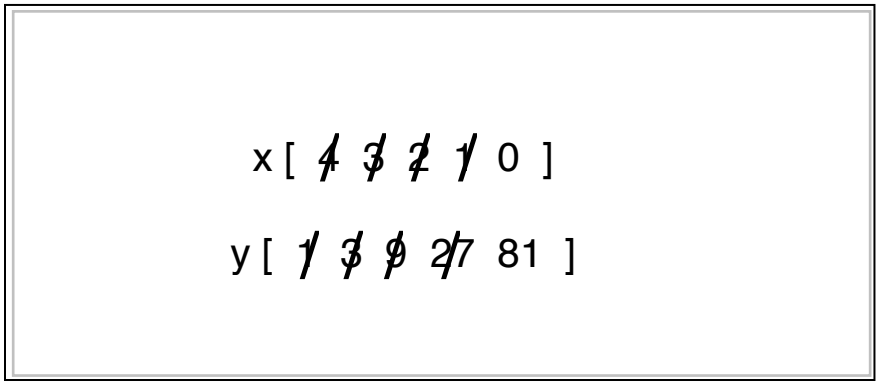
- Create the image using the shapes in powerpoint
- Use Windows' "Snipping Tool" to capture a rectangle including your diagram
- Save the Snipping Tool file to your hard drive.
- Click on the picture control in this document.
- Pick the file containing your image

Make sure you scale your image so it fits in the box you are given.

```
int x;
int y;
x = 42;
y = x + 2;
x = 46;
```




```
int x = 4;
int y = 1;
while (x > 0)
{
    y = y * 3;
    x = x - 1;
}
```



Code

Eclipse has a feature named “scrapbook” that lets you try out some code to see it does without needing the surrounding code like you saw in this lab. Let’s use that to practice a little bit. To create a scrapbook, right click on your project and pick New -> Other -> Java Run/Debug -> Scrapbook. That will bring up an empty sheet in your editor. You can write any code you want there. When you want to run the code, highlight the snippet you want to run. Right click on it and select Execute. Any output from the code will be shown in the console window.

For the problems in this part of the assignment, use the scrapbook to write the code and then copy/paste it here when it works.

1. Write a snippet that declares an integer variable named x, gives x the value 42, and then has this statement: System.out.println(“The secret to life is “ + x);

Paste your code here:

```
int x = 42;
System.out.print("The secret to life is " + x);
```

Paste the output of your code here:

```
The secret to life is 42
```

2. Write a loop that will output the numbers from 1 to 10 each on its own line.

Paste your code here:

```
int x = 1;
while (x < 11)
{
	System.out.print(x);
	x = x + 1;
	System.out.println(" ");
}
```