

Name: Andrew Januszko

CSC110 Exam 1

Vocabulary - 1 pt each

Use the following words to fill in the blanks in this section. Note: you can use words more than once, you may need to pluralize the words, and you will not use all of the words

Algorithm	Assembly language	BigInteger	Compiler
Constant	Constructor	double	Getter
High-level language	Interpreted	Keyword	Mantissa
Method	Naming Convention	null	Object
Parameter	Precedence	Primitive type	Return Type
Runnable	String	Target Machine	Typecast

1. A class with a main method is Runnable.
2. A sequence of characters is called a String.
3. If the compiler knows how much space a type requires, that is a primitive type.
4. The output of a compiler is meant for a specific Target Machine.
5. Objects get created by methods called Constructors.
6. You typecast a value when you want to force it to be a different type.
7. A method that does not return a value has a return type of void.
8. When you are being the machine and evaluating an expression, it is import to pay attention to the order of precedence of operations to know which one to do first.
9. We pass information into methods using parameters.
10. When we make variable names, we can't use Keywords because the compiler knows about them and has already given them a meaning.
11. An Assembly language is specific to one target machine while a High level language can be compiled to run on many different target machines
12. The Math.round() method has one method which lets you pass in a double.

Short Answer

13. (2 points) What is the difference between source code and object code?

Source code is higher level code that contains comments that the computer will ignore.
Object code is code designed to run on a specific computer.

14. (4 points) In your own words, explain this sentence, "The machine is short sighted and sequential."

The machine only remembers what it needs at that time and forgets it if it no longer needs it. It also works line by line instead of running code as a whole.

15. (2 points) Give the range of numbers that will be generated by

`(int)(Math.random() * 5 + 13)`
min max **[8, 13)**

16. (4 points) What is the difference between interpreted languages and compiled languages

Interpreted is translated as the computer executes it and compiled is already translated when the computer runs it.

17. (4 points) Explain the difference between step into and step over.

Step into steps into a method and only runs until the breakpoint.
Step over steps over a method after executing it.

18. (1 points each) What type of statement do you use for each of these situations:

(a) to create a variable `int x`

(b) to give a variable a value `x = 4;`

(c) to choose to do something or not do it `if { } else { }`

(d) to do something more than once `while (x < 6) { } 3`

(e) to allocate space `public Book { }`

19. (2 points) How do we read an assignment statement and why do we not use the word "equals"?

Because it is not equal to that value, it is just assigned that value.

20. (2 points) What is an instance variable?

A variable that is created that every method gets a copy of "private int x"

Java code template
for creating instance
variables of that type

4 setter related comments
4 type we are dealing

Code Constructs

21. (4 points) Write the code to declare a variable named var1 that can hold a real number and give it the value 42.3

```
double var1 = 42.3;
```

22. (4 points) Write the for loop that is equivalent to this while loop

```
int x;  
x = 5;  
while (x < 10)  
{  
    System.out.println(x);  
    x++;  
}
```

```
for (int x = 5; x < 10; x++) {  
    System.out.println(x);  
}
```

23. (4 points) Show the declaration of a method named OhMy that returns a real value and has two parameters: a string and an integer

```
public OhMy (String n, int c) {
```

24. (3 points) How do we call a constructor to create a new object of a class we have declared?

```
Using:  $\{ \}$   
<constructor> <name> = new <constructor> (<param>)
```

25. (2 points) How do we know a variable is an instance variable?

When it is declared at the beginning of the class

26. (2 points) How do you know a method is a constructor?

When its name matches the class name

Be The Machine

27. (2 points) Draw the memory diagram for the following code snippet

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

y[44]

x[42 46]

28. (4 points) Draw the memory diagram for the following code snippet

```
double[] x = new double[5];  
x[3] = 42;
```

x[0, 0, 0, 42, 0]

29. (3 points) What is the output from this code snippet?

```
int x;  
x = 42;  
if (x >= 42)  
{  
    System.out.println("Life, the Universe, and Everything");  
}  
else  
{  
    System.out.println("Not enough time")  
}
```

x[42]

Output

Life, the Universe, and Everything

30. (4 points) What is the output from this code snippet?

```
for (int i = 3; i <= 5; i++)  
{
```

```
    for (int k = 4; k > 0; k = k - 2)  
    {
```

```
        System.out.print(i + ", " + k);  
        if (i == k)
```

```
        {
```

```
            System.out.println("    SAME!!!!");
```

```
        }
```

```
        System.out.println();  
    }
```

```
}
```

```
}
```

i [3 4 5 6]

k [4 2 0 4 2 0 4 2 0]

Output

3, 4

3, 2

4, 4 SAME!!!!

4, 2

5, 4

5, 2

31. (1 points each) What is the value of each of these expressions

(a) $13 + 3 * 4$ 25

(b) $5\%2$ 1

(c) $10\%15$ 15

(d) $3\%3$ 0

32. (4 points) Suppose you have written a class named MyClass whose toString method looks like this:

```
public String toString()  
{
```

```
    return "This is one fancy object";
```

```
}
```

What would be the output from this code snippet?

```
MyClass x = new MyClass();
```

```
System.out.println("I really like " + x);
```

I really like This is one fancy object

For the next few questions, suppose we have the following class declaration

```
public class PerfectThing
{
    private int clean;

    public PerfectThing(int hold)
    {
        clean = hold;
    }

    public int getTheMagic()
    {
        return hold * 42;
    }

    public void setClean(int nextClean)
    {
        clean = nextClean;
    }
}
```

Draw the memory diagram for each of these code snippets

33. (4 points)

```
PerfectThing x = new PerfectThing(2);
```

x [84]

hold [2]

34. (4 points)

```
PerfectThing x = new PerfectThing(2);
PerfectThing y;
y = x;
y.setPerfectThing(-1);
```

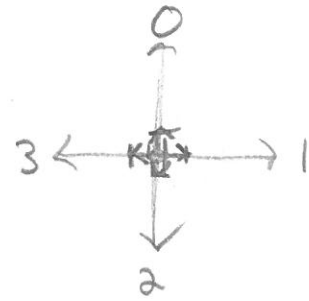
x [84]

y [84, -42]

Rogue

These declarations are at the top of our Player class:

```
static final int NORTH = 0;
static final int EAST = 1;
static final int SOUTH = 2;
static final int WEST = 3;
static final int IDLE = 4;
static final int[] CHANGE_IN_X = { 0, 1, 0, -1 };
static final int[] CHANGE_IN_Y = { -1, 0, 1, 0 };
```



1. (2 points) How do the direction constants relate to the CHANGE_IN_X constant?

They are set values to describe where the player is going

2. (2 points) We had an instance variable for the player's current x position (x) and the direction the player wants to move in (direction). How do we calculate the player's next x position?

by adding the change in the x and y and then setting a new variable accordingly

3. (5 BONUS points!) How could we change CHANGE_IN_X so that, when we are calculating the next position, we don't have to do something special if the player is idle?

(x) ?