1. For each of the parts below, give a short answer (a few words, perhaps a sentence or two).

(a) What does it mean to “cast” an expression to a type?

(b) Explain why the following code snippet needs to have some curly-braces {} added:

```java
if (x == y)
    doThis();
    doThat();
```

(c) Explain what is wrong with this `main()` method, and how to fix it:

```java
public static void main(String[] args)
{
    if (args.length == 0)
    {
        System.out.println("ERROR: You must give some arguments!");
    }

    System.out.println("This is your word: "+args[0]);
}
```

(d) Explain the difference between ‘a’ and "a".

(e) The operator + can be used for two types of operations: addition, and one other. Name the other, and explain how Java knows which of the two operations to perform.
(f) What is the type returned by the following expression?
\[ x == y \]

(g) Suppose that you have an \texttt{int} variable named \texttt{x}, and it is set to the largest value that an \texttt{int} can hold. What is the value of \texttt{y} after this line of code?
\[
\text{int } y = x+1;
\]

(h) Given your experience with \texttt{Integer.parseInt()}, you can fill in most of its declaration. Fill in the blanks in the code below:
\[
\text{public static } \texttt{integer} \quad \texttt{integer}(\texttt{integer} \ \texttt{input})
\]

(i) Consider the following line of code:
\[
\text{String}[] \ \text{array} = \text{new String}[10];
\]

How many \texttt{String} objects have been allocated, and what are their contents?
2. Select an appropriate type for each of the following types of information. Instead of using the “good defaults,” choose the smallest type which can reasonably store the information.

You may use primitives, String, and arrays.

(a) The number of cars in a parking lot

(b) The world record time (in seconds) for an Olympic event

(c) The size of an atom, in meters

(d) Whether or not a student was in class today

(e) The attendance records for the entire class today

(f) The first letter of a word

(g) The name of a company

(h) The capacity of a hard disk, in bytes
3. Give the type and value of each of the following expressions:

\[ 10 \times 3 \quad / \quad 3 \]

\[ 10 \times 3 \quad / \quad 3.0 \]

\[ 10 \times 3.0 \quad / \quad 3.0 \]

\[ 10 \quad / \quad 3 \quad \times \quad 3 \]

\[ 10 \quad / \quad 3 \quad \times \quad 3.0 \]

4. What is printed by the method `first()` below?

```java
public static void first()
{
    int x = 100;
    int y = second(x);
    int z = third(x);

    System.out.println("x="+x);
    System.out.println("y="+y);
    System.out.println("z="+z);
}

class Test
{
    public static int second(int val)
    {
        return (val+200)/val;
    }

    public static int third(int val)
    {
        if (val != 100)
            return 17;
        else
            return 23;
    }
}
```
5. Examine the following line of code. Explain exactly what methods are called, and in what order. Be clear when each method returns, and when the next begins:

   System.out.println(foo(bar(baz())));

6. Suppose that a String variable’s value is null. Give an example of something you can do with the variable which will not result in a NullPointerException (other than simply setting it to some other value). Then give an example of something you can do which will compile, but which will result in a NullPointerException at runtime.
7. Write a method which takes a single parameter, which is an array of `int`, and returns the sum of all of the **even** elements.

8. Write a method which takes two arrays of `int` as parameters, and returns true if both have exactly the same contents (including the same length).
9. Write a method which replicates `toCharArray()`: it takes a single `String` as input, and returns an array of characters, which are the characters from the string. The method may not call `toCharArray()`, but is allowed to call any of the other methods we have discussed in class, including:

- `length()`
- `equals()`
- `compareTo()`
- `charAt()`
- `substring()`
- `indexOf()`

(You will not need most of these methods.)
10. Write a class which has three instance fields and two instance methods. You can decide what the fields are. For the methods, one of them must return `void` and change the state of the class in some way; the other must not change the state of the class at all, but must return some value calculated from the fields.

Remember that all fields and methods must be `public`. 