3.2)

{1, 3, 5, 7, 9, 9, 7, 5, 3, 1}

3.3)

for (int negatives : list)

{

4.1)

FileNotFoundException occurs and the program stops. The JVM will then prompt the user for the next move.

4.2)

If you use PrintWriter to write a file and the file doesn’t exist then a new one will be created.

5.1)

When an exception is thrown the program will terminate and the exception will be printed. When an exception is caught the programer has the opportunity to address the exception with a user friendly interface that tells the user what happened. The programmer can then add any more code to prompt the user to try again.

5.2)

A checked exception is an exception that is outside the programmers control. One example is the FileNotFoundException which means the user or some other factor outside the program has yet to enter a proper file name.

5.3)

An unchecked exception is an error in the code. The programmer is at fault for this type of error. One example of this is the IllegalArgumentException. Let’s say a method had a return value of int and in the code a double was returned. This would signify an IllegalArgumentException.

5.4)

All throwable exceptions can be declared after the throws reserved word.

5.5)

If an IndexOutOfBounds exception occurs the JVM will output the prompt and stacktrace for the exception while simultaneously stopping the program.

5.6)

It is not always the same because in a catch you can prompt the program to output another exception.

6.3)

Instance methods are invoked on an object and static methods do not. An instance method will have an object before the method call (i.e. object.method();). A static method will call a method without an object (i.e. method()). It is hard to tell the difference between calls to instance and static methods from inside an instance method because inside an instance method there can be calls made to both methods without needing an object reference.

6.4)

When this method is called there is a series of syntax errors. First pArgs in the main method does not exist. A runtime error occurs when String s is not initialized with a value and the constructor calls on x = s.length(); for an object that is null. The main method is calling for a new constructor method but it isn’t attaching it to any object.

6.8)

1. This will not compile because it’s calling on a private variable from outside the method.
2. This will not compile for the same reason.
3. Sam with this one.
4. This one will compile because the variable is public.
5. This won’t compile because the object is calling the class to execute the method.
6. This one compiles because the object returns an integer from a public method that is instantiated by a constructor.
7. This one will execute the command and set mX to 20. It passes an implicit parameter to the public setX method.
8. This will compile and what is executed when it happens is ???
9. This will not compile because H1\_65 is a class and it is asking a class to return a variable instead of attaching it to an object.
10. This method will not compile because the instance method inside of the class is not static.
11. This will work because the integer value that is returned when cObj1 is returned is an integer.
12. This method will work because it uses and object to call an instance method and passes a value to a setter method.
13. This won’t work because it’s calling on the class to return an instance method that is not static. If it used an object to call the method it would work.
14. This one is also calling on the using the class to reach a method when it needs to use an object to reach an instance method.