Name: _____

1. (10 points) For the following, Check T if the statement is true, the F if the statement is false.

   (a) ☐ T ☐ F : In mathematics, we learned that every integer, n has a successor, $n + 1$, that is greater than n. This is also true in Java.

   (b) ☐ T ☐ F : There are four different types of integers in Java so that we can save memory.

   (c) ☐ T ☐ F : The left hand side of an assignment statement can be a literal constant.

   (d) ☐ T ☐ F : A reference variable does not have to refer to an object, but in order to use an object, it must be referenced by a reference variable or parameter.

   (e) ☐ T ☐ F : In Java, some methods are not in any class.

   (f) ☐ T ☐ F : A single Java code file, compiled on different machines, can produce different results.

   (g) ☐ T ☐ F : If aliens from outer space prevent all computers from typing (or cut and pasting) the character "}" (right curly brace), then no one could write any new Java code.

   (h) ☐ T ☐ F : Multiple Java files can be in a single package, but a single Java file cannot be in multiple Java packages.

   (i) ☐ T ☐ F : There are eight primitive types in java char, bytes, short, long, float, double, and quad.

   (j) ☐ T ☐ F : An Array in Java an object with a length field, a field for each element of the array, and a toString field.

2. (10 points) For the following, put a check in front of the statements that are are valid constant expressions.

   ☐ 32_768
   ☐ 9934
   ☐ 0x1ABd
   ☐ 6.23e22
   ☐ 'so called'
   ☐ 0340721
   ☐ 'x'
   ☐ 0b729
   ☐ "CS-140"
   ☐ 4.729

3. (10 points) For the following, put a check in front of the statements that are valid declaration statements.

   ☐ int x=17;
   ☐ int i = 37_295;
   ☐ int j=1,024;
   ☐ byte small = 3 * 110;
   ☐ float ____pi=3.1514;
   ☐ final double IF = 6.79e3;
   ☐ double break=5.0;
   ☐ final double IF = 6.79e3;
   ☐ double bigNumber=inf;
   ☐ int num_cookies=093;

4. (10 points) For the following, put a check in front of the statements that are syntactically correct, assuming the following declare statements:

```
int x=3; char init='T';
```

- ☐ x = /* 3 / 0 */ x + 2;
- ☐ x = /* /* 3 / 0 */ x / 0 */ * 3;
- ☐ x = (int)init + 3;
- ☐ init = x+3;
- ☐ x = x % 12;
- ☐ init = / 'Z';
- ☐ init = init;
- ☐ x = 4.3 / 1.765 ;
- ☐ x = 21 if (init=='T');
- ☐ x*3=9;

5. (10 points) Given the following declarations:

```
int a = 13; int b=4; int c=2; double fx=1.3; double fy=-7.0;
```

Determine the value of the following expressions (Be sure to use a decimal point in floating point answers, and leave out decimal points in integer answers.)

(a) _____ = a + b

(b) _____ = a % c

(c) _____ = (fy >= 0) ? 'p' : 'n'

(d) _____ = b + 17

(e) _____ = fy / b

(f) _____ = fy + 7

(g) _____ = (a + fx) * 2

(h) _____ = (a<0) && (b!=1)

(i) _____ = fy < fx

(j) _____ = 12 - 8.0

6. (10 points) Given the following Java program:

```
package test1;
public class Tester {
   public static void main(String[] args) {
      System.out.println("Testing the Location class");
      Location point1=new Location(10,10);
      System.out.println(" point1 - Expect (10,10): " + point1);
      Location point2 = new Location(point1);
      System.out.println(" point2 - Expect (10,10): " + point2);
      System.out.println(" point1.equals(point2) - Expect true: " +
         point1.equals(point2));
      System.out.println(" point1.inBounds() -  Expect true: " +
         point1.inBounds());
   }
}
```

(a) How many instance methods are in this code? _____

(b) How many class methods are in this code? _____

(c) Is "Location" a constructor or an instance method? How can you tell?

_____

(d) Is "inbounds" a constructor or an instance method? How can you tell?

_____

(e) When the code invokes "Location(75,75)", will it run the same code as when it invoke "Location(point1)"? Why?

_____

(f) What are the advantages of testing the Location class from a different class, instead of testing the code in the Location class itself?

_____

7. (10 points) The following is the contents of a file in the current directory called "AvgArgs.java":

```
package test01;

public class AvgArgs {

   public static double avgArgs(String[] args) {
      double sum=0;
      for(String arg: args) { sum += arg.length(); }
      return sum/args.length;
   }

   public static void main(String[] args) {
      System.out.println("Average argument length: " + avgArgs(args));
   }
}
```

If you type the command: **javac -d . AvgArgs.java**, you will get no errors. If you then type the command: **java -cp . test01.AvgArgs This is a test**

What will get printed to the monitor?

8. (10 points) Given the following code:

```
package test01;
public class Account {
   private double balance; private double interestRate; private int year;
   public Account(double initBalance,double initRate) {
      balance=initBalance; interestRate=initRate; }
   public double getBalance() { return balance; }
   public double getInterestRate() { return interestRate; }
   public void setInterestRate(double newRate) { interestRate=newRate; }
   public int getYear( ) { return year; }
   public void deposit(double dep) { balance += dep; }
   public double withdraw(double req) {
      if (req > balance) { req=balance; }
      balance -=req; return req; }
   public double waitYear() {
      double interest = balance * interestRate;
      this.deposit(interest); year++; return interest; }
   public String toString() { return "Year: " + year + " - $" + balance; }
   public static void main(String[] args) {
      Account test = new Account(1000.0,0.05);
      System.out.println(test);
      test.deposit(85.63); test.waitYear(); System.out.println(test);
      test.withdraw(12.99); test.deposit(100.00); test.waitYear();
      System.out.println(test);
      test.withdraw(12.99); test.deposit(259.00); test.waitYear();
      System.out.println(test);
   }
}
```

(a) An invocation record is pushed onto the program stack every time a method invokes another method in Java. During execution of the main method on the above code, what is the maximum depth of the program stack, and which methods are on the program stack? (You may assume that library methods do not call any lower level methods)

(b) Does the main method test all the other methods in the Account class? If not, which methods are not tested?

(c) Is the Account class an immutable class? If so, why, or if not, why not?

(d) Name all class methods in the Account class.

---

9. (20 points) Find as many bugs as you can in the following code. (There are between 5 and 15 bugs in the code.) A bug may cause a compile error, or may compile but produce incorrect results. If you leave out a keyword that is the default anyway, that is not a bug. For each bug, specify the line number, the incorrect code, and your best guess at what the correct value should be. If the bug is caused by missing code, enter the line number that should precede the missing code, followed by a plus sign (+), and enter the word "missing" in the Error column.

```
1  public class Buggy {
2     private int[] data;
3     public Buggy(nums...) {
4        for ( int i=0; i<nums.length(); i++)  {
5           data[i] = nums[i];
6     }
7     public getData() { return data; }
8     public double average() {
9        int sum;
10       for(i : data) { sum+=data[i]; }
11       return sum/data.length;
12    }
13 }
```

Bug	Line	Error	Correction
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			