1. Objects contain *data* and what else?
   methods

2. What does the word **class** imply?
   a collection of some category

3. List 7 **Math** class methods.
   abs, pow, sqrt, round, floor, ceil, min, max

4. List 2 **Math** class data members.
   PI and E

5. Methods of the **Math** class are called what type of methods?
   Class Methods

6. What is wrong with program **Java0601.java**?
   The Bank class does not allow access in the same manner as the Math class.

7. Refer to the previous question. How does program **Java0602.java** fix the problem?
   You need to create an object of the Bank class to have proper access.

   Refer to program **Java0602.java** for questions 8 and 9.

8. Refer to program **Java0602.java**. In the statement `Bank tom;` what is Bank and what is tom?
   Bank is the class and tom is the object.

9. What does the statement `tom = new Bank();` do?
   It instantiates a tom object of the Bank class

10. For right now, the author wants you to think of constructing an object as a combination of what 2 things?
    Declaring a variable to be an object or a class and instantiating the object

    Refer to program **Java0603.java** for questions 11 through 14.

11. What **Bank** class method makes the checking and savings accounts start at $0.00 when a new Bank object is created?
    constructor

12. Print 2 lines of Java code that will create a Bank object for yourself and initialize both your checking and your savings account to $1 million.
    `Bank myAccount;
    myAccount = new Bank(1000000, 1000000);`
13. Print 1 line of Java code that will make a $250,000 deposit in your checking account.
   `myAccount.changeChecking(250000);`

14. Print 1 line of Java code that will make a $500,000 withdrawal from your savings account.
   `myAccount.changeSavings(-500000);`

   Refer to program `Java0604.java` for questions 15 and 16.

15. What action is performed by the `getCombined` method?
   *It returns the sum of the checking and savings account balances.*

16. What actions are performed by the `closeChecking` and `closeSavings` methods?
   *It changes both account balances to zero.*

17. At the top of this hierarchy is the __________ package.
   `java`

18. The most important package is the __________ package.
   `java.lang`

19. Refer to the previous question. Do you have to import this package?
   *No, it is automatically loaded*

20. In the statement `import java.util.Random;` what is the class and what is the package?
   *The package is `util` and the class is `Random`*

21. All package identifiers start with a __________ letter and a class identifier starts with a __________ letter.
   *`lower-case` `upper-case`*

22. What does the `nextInt` method of the `Random` class do?
   *It returns a random integer*

23. Look at programs `Java0605.java` and `Java0606.java`. Why does the execution of the former produce different results each time while the execution of the latter produces the same results each time?
   *Output is different with the default constructor. With a parameter constructor the seed value makes each output sequence the same*

24. Methods with multiple capabilities are called ____________.
   *Overloaded*

25. Speaking algebraically if `n` is the integer parameter of the `nextInt` method.
   What is the largest number you can get in return?
   *`n-1`*

26. In addition to the `Random` constructor, what other method is used to set the seed for computing random numbers?
   *`setSeed`*
27. What method is used to produce random real numbers between 0 and 1?
   nextDouble

Assume that \texttt{rand} is an object of the \texttt{Random} class for questions 28 through 35.

28. What range of numbers can be displayed with: \texttt{System.out.println(rand.nextInt(90) + 10);} ?
   10 ... 99

29. What range of numbers can be displayed with: \texttt{System.out.println(rand.nextInt(50) + 1);} ?
   1 ... 50

30. What range of numbers can be displayed with: \texttt{System.out.println(rand.nextInt(120) + 60);} ?
   60 ... 179

31. What range of numbers can be displayed with: \texttt{System.out.println(rand.nextInt(100));} ?
   0 ... 99

In questions 31 - 35, fill in the missing parameters to obtain the desired range of random #s.

32. Range [1..100] \texttt{System.out.println(rand.nextInt(_100______) + _1______);}

33. Range [0..100] \texttt{System.out.println(rand.nextInt(_101______) + ____________);}

34. Range [20..60] \texttt{System.out.println(rand.nextInt(_41______) + _20______);}

35. Range [100..999] \texttt{System.out.println(rand.nextInt(_900______) + _100______);}

36. Look at program \texttt{Java0611.java}. How does this program manage to display random characters?
   \texttt{It generates random integers that are equated to ASC character values.}

For questions 37 through 40, rewrite the \texttt{println} statements from questions 32 through 35 so that they use \texttt{Math.random}.

37. Rewrite #32: \texttt{System.out.println( (int) (Math.random() * 100) + 1);}  

38. Rewrite #33: \texttt{System.out.println( (int) (Math.random() * 101) + 0);} // + 0 is optional

39. Rewrite #34: \texttt{System.out.println( (int) (Math.random() * 41) + 20);}  

40. Rewrite #35: \texttt{System.out.println( (int) (Math.random() * 900) + 100);}