|  |  |  |
| --- | --- | --- |
| **APCS Exposure Java** | **Exercises 06.01-05** | **Date:** |
| **Name:** | | **Period:** |

1. Objects contain *data* and what else?

2. What does the word **class** imply?

3. List 7 **Math** class methods.

4. List 2 **Math** class data members.

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

6. What is wrong with program ***Java0601.java***?

7. Refer to the previous question. How does program ***Java0602.java*** fix the problem?

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*?

9. What does the statement **tom = new Bank();** do?

10. For right now, the author wants you to think of *constructing an object* as a combination of what 2 things?

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?

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.

13. Print 1 line of Java code that will make a $250,000 deposit in your checking account.

14. Print 1 line of Java code that will make a $500,000 withdrawal from your savings account.

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

15. What action is performed by the **getCombined** method?

16. What actions are performed by the **closeChecking**and**closeSavings** methods?

17. At the top of this hierarchy is the **\_\_\_\_\_\_\_\_\_\_** package.

18. The most important package is the **\_\_\_\_\_\_\_\_\_\_\_\_\_**package.

19. Refer to the previous question. Do you have to import this package?

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

21. All package identifiers start with a \_\_\_\_\_\_\_\_\_\_ letter and a class identifier starts with a \_\_\_\_\_\_\_\_\_\_ letter.

22. What does the **nextInt** method of the **Random** class do?

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?

24. Methods with multiple capabilities are called \_\_\_\_\_\_\_\_\_\_\_\_\_.

25. Speaking *algebraically* if **n** is the integer parameter of the **nextInt** method.

What is the largest number you can get in return?

26. In addition to the **Random** constructor, what other *method* is used to set the seed for computing random numbers?

27. What method is used to produce random real numbers between 0 and 1?

Assume that **rand** is an object of the **Random** class for questions 28 through 35.

28. What range of numbers can be displayed with: **System.out.println(rand.nextInt(90) + 10);** ?

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

30. What range of numbers can be displayed with: **System.out.println(rand.nextInt(120) + 60);** ?

31. What range of numbers can be displayed with: **System.out.println(rand.nextInt(100));** ?

In questions 31 through 35,

fill in the missing parameters to obtain the desired range of **Random** numbers.

32. Range [1..100] **System.out.println(rand.nextInt(\_\_\_\_\_\_\_\_\_\_) + \_\_\_\_\_\_\_\_\_\_);**

33. Range [0..100] **System.out.println(rand.nextInt(\_\_\_\_\_\_\_\_\_\_) + \_\_\_\_\_\_\_\_\_\_);**

34. Range [20..60] **System.out.println(rand.nextInt(\_\_\_\_\_\_\_\_\_\_) + \_\_\_\_\_\_\_\_\_\_);**

35. Range [100..999] **System.out.println(rand.nextInt(\_\_\_\_\_\_\_\_\_\_) + \_\_\_\_\_\_\_\_\_\_);**

36. Look at program ***Java0611.java***. How does this program manage to display random *characters*?

For questions 37 through 40, rewrite the **println** statements

from questions 32 through 35 so that they use **Math.random**.

37 Rewrite #32: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

38. Rewrite #33: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

39. Rewrite #34: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

40. Rewrite #35: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_