**Chapter 8 Creating Object Methods**

01. Which Java keyword is required to create your own class methods?

**static**

02. Which Java keyword is required to allow access to object methods?

**new**

03. Class methods are typically used when

only a single copy of the class needs to be loaded

04. Object methods are typically used when

multiple copies or instances of a class are required.

05. Show correct syntax to create an object of the **Piggy** class?

**Piggy tom = new Piggy();**

06. Define an object

one instance of a class.

07. Calling a class method requires using what syntax. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_

a class identifier followed by a dot and a method identifier.

08. Calling an object method requires using\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_

an object identifier followed by a dot and a method identifier.

09. InitDatamethods and constructor methods both initialize object data. Which of these two methods gives greater program reliability and why?

Constructor methods, because they are called automatically.

10. Which features can you use to recognize constructor methods in a class declaration?

Both A and C

11. When is a constructor called?

During the instantiation of a new object

12. A class can have

multiple constructors with the same identifier.

13. What is an *overloaded* constructor?

A second or other multiple constructor with a different signature than any other constructor.

14. Constructor methods are

not required to be placed at any particular position in a class declaration.

15. Which of the following method declarations can be a constructor?

I. **public static void Qwerty()**

**{**

**start = 0;**

**max = 1000;**

**}**

II. **public Qwerty()**

**{**

**start = 0;**

**max = 1000;**

**}**

III. **public Qwerty(int s, int m)**

**{**

**start = s;**

**max = m;**

**}**

II & III only

16. Which of the following statements calls a constructor method where **Qwerty** is a class identifier?

**Qwerty balloon = new Qwerty();**

17. Without the use of a **public** or **private** declaration, access to data and methods is

available to methods of the same class and other classes.

18. Access to **private** data or **private** methods is

restricted to methods of the same class.

19. Access to **public** data or **public** methods is

available to methods of the same class and other classes.

20. The use of **private** in a class declaration

creates greater program reliability.

21. Consider the program below.

**public class Waco**

**{**

**public static void main (String args[ ])**

**{**

**Piggy kathy = new Piggy("Kathy",1500.0);**

**Piggy rachel = new Piggy("Rachel",2500.0);**

**kathy.showData(); // Line 1**

**System.out.println("Name " + rachel.name); // Line 2**

**System.out.println("Savings " + rachel.savings); // Line 3**

**}**

**}**

**class Piggy**

**{**

**public double savings;**

**public String name;**

**public Piggy(String n, double s)**

**{**

**name = n;**

**savings = s;**

**}**

**public void showData()**

**{**

**System.out.println("Name: " + name); // Line 4**

**System.out.println("Savings: " + savings); // Line 5**

**}**

**}**

Lines 1 - 5 access data of **Piggy** objects. Which lines have access?

All five lines have access

22. Consider the program below.

**public class Waco**

**{**

**public static void main (String args[ ])**

**{**

**Piggy kathy = new Piggy("Kathy",1500.0);**

**Piggy rachel = new Piggy("Rachel",2500.0);**

**kathy.showData(); // Line 1**

**System.out.println("Name " + rachel.name); // Line 2**

**System.out.println("Savings " + rachel.savings); // Line 3**

**}**

**}**

**class Piggy**

**{**

**private double savings;**

**public String name;**

**public Piggy(String n, double s)**

**{**

**name = n;**

**savings = s;**

**}**

**public void showData()**

**{**

**System.out.println("Name: " + name); // Line 4**

**System.out.println("Savings: " + savings); // Line 5**

**}**

**}**

Lines 1, 2, 4 and 5 only

23. Consider the program below.

**public class Waco**

**{**

**public static void main (String args[ ])**

**{**

**Piggy kathy = new Piggy("Kathy",1500.0);**

**Piggy rachel = new Piggy("Rachel",2500.0);**

**kathy.showData(); // Line 1**

**System.out.println("Name " + rachel.name); // Line 2**

**System.out.println("Savings " + rachel.savings); // Line 3**

**}**

**}**

**class Piggy**

**{**

**public double savings;**

**private String name;**

**public Piggy(String n, double s)**

**{**

**name = n;**

**savings = s;**

**}**

**public void ShowData()**

**{**

**System.out.println("Name: " + name); // Line 4**

**System.out.println("Savings: " + savings); // Line 5**

**}**

**}**

Lines 1, 3, 4 and 5 only

24. Consider the program below.

**public class Waco**

**{**

**public static void main (String args[ ])**

**{**

**Piggy kathy = new Piggy("Kathy",1500.0);**

**Piggy rachel = new Piggy("Rachel",2500.0);**

**kathy.showData(); // Line 1**

**System.out.println("Name " + rachel.name); // Line 2**

**System.out.println("Savings " + rachel.savings); // Line 3**

**}**

**}**

**class Piggy**

**{**

**private double savings;**

**private String name;**

**public Piggy(String n, double s)**

**{**

**name = n;**

**savings = s;**

**}**

**private void showData()**

**{**

**System.out.println("Name: " + name); // Line 4**

**System.out.println("Savings: " + savings); // Line 5**

**}**

**}**

Lines 1 - 5 access data of **Piggy** objects. Which line(s) have access?

Lines 4 and 5 only

25. Which of the following is the minimum class declaration that will compile?

**class CardDeck**

**{**

**}**

26. The constructor is used to

initialize class data and call other methods if they are necessary to construct a new object.

27. Which class members should be declared as **private**?

Predominantly data attributes and some helper methods

28. Which class members should be declared as **public**?

Methods only and occasionally attributes

29. A **class** method

requires using the keyword **static**.

30. An **object** method

requires using the keyword **new**.

31. A **private** method

I. can only be accessed by methods of the same class.

II. is usually a helper method.

III. can never be a constructor.

I, II & III

32. A **public** method

I. can only access public data.

II. can be a constructor.

III. can be accessed from outside the class.

II & III only

33. A **void** methodcan also be a(n)

All of the above

34. A **return** method

A & B only

35. A **default** constructor is a

*no-parameter* method, which is called automatically during the instantiation of a new object.

36. A **parameter** constructor is a

*parameter* method, which is called automatically during the instantiation of a new object.