Chapter Number: 9 Inheritance

Question type: Multiple Choice

1. A class that represents the most general entity in an inheritance hierarchy is called a/an \_\_\_\_.

a) Default class.

b) Superclass.

c) Subclass.

d) Inheritance class.

Answer: b

Title: The most general class in an inheritance hierarchy is called \_\_\_.

Difficulty: Medium

Section Reference: 9.1 Inheritance Hierarchies

2. A class that represents a more specific entity in an inheritance hierarchy is called a/an \_\_\_\_.

a) Default class

b) Superclass

c) Subclass.

d) Inheritance class.

Answer: c

Title: A more specific class in an inheritance hierarchy is called \_\_\_\_.

Difficulty: Medium

Section Reference: 9.1 Inheritance Hierarchies

3. You are creating a class inheritance hierarchy about motor vehicles that will contain classes named Vehicle, Auto, and Motorcycle. Which of the following statements is correct?

a) Vehicle should be the default class, while Auto and Motorcycle should be the subclasses.

b) Vehicle should be the superclass, while Auto and Motorcycle should be the subclasses.

c) Vehicle should be the subclass, while Auto and Motorcycle should be the superclasses.

d) Vehicle should be the subclass, while Auto and Motorcycle should be the default classes.

Answer: b

Title: Creating a class inheritance hierarchy

Difficulty: Medium

Section Reference: 9.1 Inheritance Hierarchies

4. Which of the following statements about inheritance is correct?

a) You can always use a superclass object in place of a subclass object.

b) You can always use a subclass object in place of a superclass object.

c) A superclass inherits data and behavior from a subclass.

d) A superclass inherits only behavior from a subclass.

Answer: b

Title: Which statement about inheritance is correct?

Difficulty: Medium

Section Reference: 9.1 Inheritance Hierarchies

5. Insert the missing code in the following code fragment. This fragment is intended to call the Vehicle class's method.

public class Vehicle

{

. . .

public void setVehicleClass(double numberAxles)

{

. . .

}

}

public class Motorcycle extends Vehicle

{

. . .

public Motorcycle()

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

}

a) Motorcyle.setVehicleClass(2.0);

b) Vehicle.setVehicleClass(2.0);

c) this;setVehicleClass(2.0);

d) setVehicleClass(2.0);

Answer: d

Title: Complete the code to call a superclass method

Difficulty: Medium

Section Reference: 9.1 Inheritance Hierarchies

6. Insert the missing code in the following code fragment. This fragment is intended to call the Vessel class's method.

public class Vessel

{

. . .

public void set VesselClass(double vesselLength)

{

. . .

}

}

public class SpeedBoat extends Vessel

{

. . .

public SpeedBoat()

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

}

a) SpeedBoat.vesselLength(26.0);

b) Vessel.vesselLength(26.0);

c) this;vesselLength(26.0);

d) vesselLength(26.0);

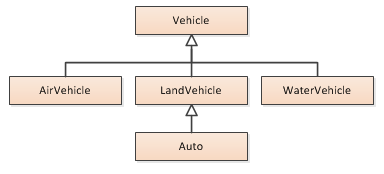
Answer: d

Title: Complete the code to call a superclass method

Difficulty: Medium

Section Reference: 9.1 Inheritance Hierarchies

7. Consider the following inheritance hierarchy diagram:



Which of the following statements is correct?

a) Auto is a superclass of LandVehicle, and LandVehicle is a superclass of Vehicle.

b) Auto is a superclass of LandVehicle, and LandVehicle is a subclass of Vehicle.

c) Auto is a subclass of LandVehicle, and LandVehicle is a superclass of Vehicle.

d) Auto is a subclass of LandVehicle, and LandVehicle is a subclass of Vehicle.

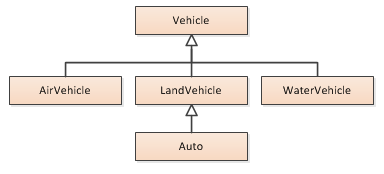
Answer: d

Title: Which statement about an inheritance hierarchy is correct?

Difficulty: Easy

Section Reference: 9.1 Inheritance Hierarchies

8. Consider the following inheritance hierarchy diagram:



Which of the following statements is correct?

a) Auto class inherits from LandVehicle class, and LandVehicle class inherits from Vehicle class.

b) Auto class inherits from LandVehicle class, and Vehicle class inherits from LandVehicle class.

c) LandVehicle class inherits from Auto class, and LandVehicle class inherits from Vehicle class.

d) LandVehicle class inherits from Auto class, and Vehicle class inherits from LandVehicle class.

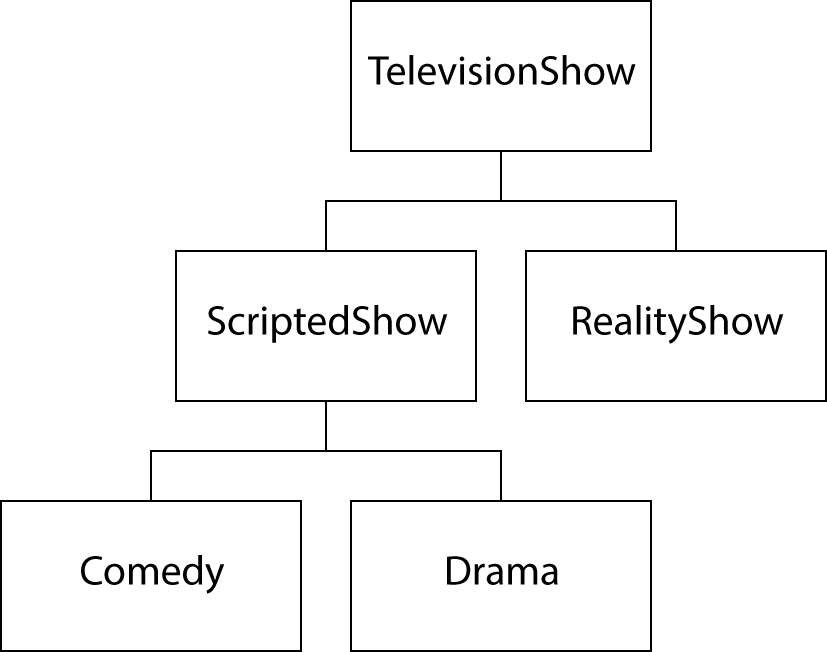
Answer: a

Title: Which statement about an inheritance hierarchy is correct?

Difficulty: Easy

Section Reference: 9.1 Inheritance Hierarchies

9. Consider the hierarchy of classes shown below.



Which represent valid class headers that would be found in this hierarchy?

a) public class ScriptedShow extends TelevisionShow {. . .  
 public class Comedy extends ScriptedShow {. . .

b) public class TelevisionShow extends ScriptedShow {. . .  
 public class ScriptedShow extends Comedy {. . .

c) public class Drama extends TelevisionShow {. . .  
 public class Comedy extends Drama {. . .

d) public class ScriptedShow extends RealityShow {. . .  
 public class RealityShow extends ScriptedShow {. . .

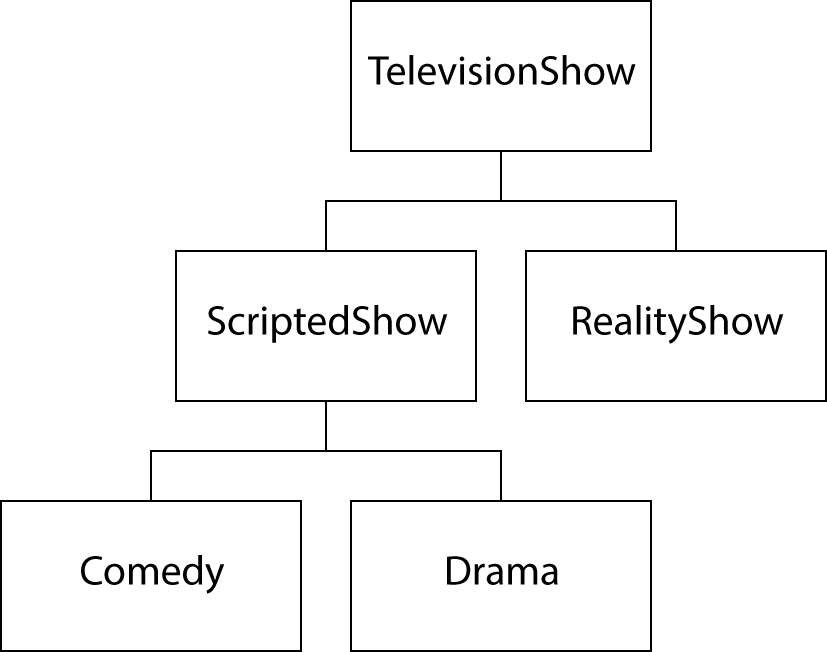
Answer: a

Title: Which represent valid class headers that would be found in this hierarchy?

Difficulty: Medium

Section Reference 1: 9.1 Inheritance Hierarchies

10. Consider the hierarchy of classes shown below.



What is the superclass of the class ScriptedShow?

a) Comedy

b) RealityShow

c) Object

d) TelevisionShow

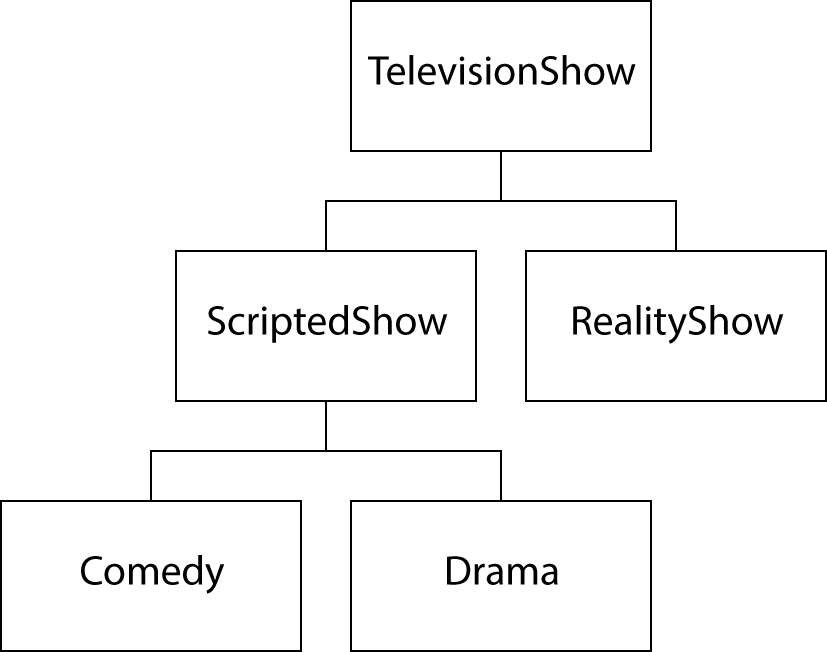
Answer: d

Title: What is the superclass of the class ScriptedShow?

Difficulty: Medium

Section Reference 1: 9.1 Inheritance Hierarchies

11. Consider the hierarchy of classes shown below.



What is the superclass of the class TelevisionShow?

a) Object

b) Comedy

c) RealityShow

d) This class has no superclass

Answer: a

Title: What is the superclass of the class TelevisionShow?

Difficulty: Medium

Section Reference 1: 9.1 Inheritance Hierarchies

12. All hamsters are rodents and all rodents are mammals. What hierarchy best captures this information?

a) Hamster is a superclass of Rodent and Rodent is a superclass of Mammal

b) Mammal is a superclass of Rodent and Rodent is a superclass of Hamster

c) Mammal is a superclass of Rodent and Hamster

d) Hamster is a superclass of Rodent and Mammal

Answer: b

Title: What hierarchy best captures this information?

Difficulty: Medium

Section Reference 1: 9.1 Inheritance Hierarchies

13. All rodents are mammals and all canines are mammals. No canines are rodents and no rodents are canines. What hierarchy best captures this information?

a) Mammal is a superclass of Rodent and Mammal

b) Rodent is a superclass of Mammal and Canine is a superclass of Mammal

c) Mammal is a superclass of Rodent and Rodent is a superclass of Canine

d) Mammal is a superclass of Canine and Canine is a superclass of Rodent

Answer: a

Title: What hierarchy best captures this information?

Difficulty: Medium

Section Reference 1: 9.1 Inheritance Hierarchies

14. Consider the classes shown below:

public class Parent

{

public void doSomething(){/\* Implementation not shown \*/}

}

public class Child extends Parent

{

public void doAnotherThing(){/\* Implementation not shown \*/}

}

Which lines in the following code will compile without error?

Child kid = new Child();

kid.doSomething(); // line 1

kid.doAnotherThing(); // line 2

a) Line 1 only

b) Line 2 only

c) Lines 1 and 2

d) Neither line will compile without error

Answer: c

Title: Which lines in the following code will compile without error?

Difficulty: Medium

Section Reference 1: 9.2 Implementing Subclasses

15. Consider the classes shown below:

public class Parent

{

public void doSomething(){/\* Implementation not shown \*/}

}

public class Child extends Parent

{

public void doAnotherThing(){/\* Implementation not shown \*/}

}

Which lines in the following code will compile without error?

Parent kid = new Child();

kid.doSomething(); // line 1

kid.doAnotherThing(); // line 2

a) Line 1 only

b) Line 2 only

c) Lines 1 and 2

d) Neither line will compile without error

Answer: a

Title: Which lines in the following code will compile without error?

Difficulty: Medium

Section Reference 1: 9.2 Implementing Subclasses

16. Consider the classes shown below:

public class Parent

{

private int value = 100;

public int getValue()

{

return value;

}

}

public class Child extends Parent

{

private int value;

public Child(int number)

{

value = number;

}

}

What is the output of the following lines of code?

Child kid = new Child(-14);

Parent adult = new Parent();

System.*out*.println(kid.getValue() + " "

+ adult.getValue());

a) 100 100

b) -14 100

c) -14 -14

d) 100 -14

Answer: a

Title: What is the output of the following lines of code?

Difficulty: Medium

Section Reference 1: 9.2 Implementing Subclasses

17. Consider the classes shown below:

public class Parent

{

private int value = 100;

public int getValue()

{

return value;

}

}

public class Child extends Parent

{

private int value;

public Child(int number)

{

value = number;

}

}

What is the output of the following lines of code?

Child kid = new Child(-14);

Child kid2 = new Child(21);

System.*out*.println(kid.getValue() + " "

+ kid2.getValue());

a) -14 21

b) 21 21

c) 21 100

d) 100 100

Answer: d

Title: What is the output of the following lines of code?

Difficulty: Medium

Section Reference 1: 9.2 Implementing Subclasses

18. Consider the classes shown below:

public class Parent

{

private int value = 100;

public int getValue()

{

return value;

}

}

public class Child extends Parent

{

private int value;

public Child(int number)

{

value = number;

}

}

What is the output of the following lines of code?

Child kid = new Child(-14);

Parent kid2 = new Child(21);

System.out.println(kid.getValue() + " "

+ kid2.getValue());

a) 100 100

b) -14 21

c) 21 21

d) -14 100

Answer: a

Title: What is the output of the following lines of code?

Difficulty: Medium

Section Reference 1: 9.2 Implementing Subclasses

19. Suppose the class Value is partially defined below

public class Value

{

private int number;

public int getValue()

{

return number;

}

}

A subclass of Value, LargerValue, is defined with a getValue method that returns twice the value of the parent. Which line is the body of LargerValue's getValue method?

a) return getValue() \* 2;

b) return super.getValue() \* 2;

c) return number \* 2;

d) return super.number \* 2;

Answer: b

Title: Which line is the body of LargerValue's getValue?

Difficulty: Medium

Section Reference 1: 9.2 Implementing Subclasses

20. Suppose the class Message is partially defined as shown below

public class Message

{

private String value;

public Message(String initial)

{

value = initial;

}

public String getMessage()

{

return value;

}

}

A subclass of Message, ExcitedMessage, is defined that will behave like Message, except that it will add two exclamation points to the end of the message. Sample code that uses ExcitedMessage is shown below.

ExcitedMessage greeting = new ExcitedMessage("Hello");

System.out.print(greeting.getMessage());

// will print "Hello!!"

Which ExcitedMessage constructor will give this behavior?

a) public ExcitedMessage(String line)  
 {  
 super(line + "!!");  
 }

b) public ExcitedMessage(String line)  
 {  
 value = line + "!!";  
 }

c) public ExcitedMessage(String line)  
 {  
 line = line + "!!";  
 super(line);  
 }

d) public ExcitedMessage(String line)  
 {  
 new Message(line + "!!");  
 }

Answer: a

Title: Which line is the body of ExcitedMessage 's constructor?

Difficulty: Medium

Section Reference 1: 9.2 Implementing Subclasses

21. To create a subclass, use the \_\_\_\_ keyword.

a) inherits

b) implements

c) interface

d) extends

Answer: d

Title: To create a subclass, use the \_\_\_\_ keyword.

Difficulty: Easy

Section Reference: 9.2 Implementing Subclasses

22. You are creating a Motorcycle class which is supposed to inherit from the Vehicle class. Which of the following class declaration statements will accomplish this?

a) public class Motorcycle inherits Vehicle

b) public class Motorcycle implements Vehicle

c) public class Motorcycle interfaces Vehicle

d) public class Motorcycle extends Vehicle

Answer: d

Title: Which class declaration statement indicates inheritance?

Difficulty: Easy

Section Reference: 9.2 Implementing Subclasses

23. You are creating a Motorcycle class which is supposed to be a subclass of the Vehicle class. Which of the following class declaration statements will accomplish this?

a) public class Motorcycle extends Vehicle

b) public class Motorcycle implements Vehicle

c) public class Motorcycle interfaces Vehicle

d) public class Motorcycle inherits Vehicle

Answer: a

Title: Which class declaration statement indicates inheritance?

Difficulty: Easy

Section Reference: 9.2 Implementing Subclasses

24. You are creating a Motorcycle class that is supposed to be a subclass of the Vehicle class. Which of the following class declaration statements will accomplish this?

a) public class Vehicle extends Motorcycle

b) public class Motorcycle extends Vehicle

c) public class Vehicle inherits Motorcycle

d) public class Motorcycle inherits Vehicle

Answer: b

Title: Which class declaration statement indicates inheritance?

Difficulty: Easy

Section Reference: 9.2 Implementing Subclasses

25. Which of the following is true regarding subclasses?

a) A subclass inherits methods from its superclass but not instance variables.

b) A subclass inherits instance variables from its superclass but not methods.

c) A subclass inherits methods and instance variables from its superclass.

d) A subclass does not inherit methods or instance variables from its superclass.

Answer: c

Title: Which is true regarding subclasses?

Difficulty: Hard

Section Reference: 9.2 Implementing Subclasses

26. Which of the following is true regarding subclasses?

a) A subclass that inherits methods from its superclass may not override the methods.

b) A subclass that inherits instance variables from its superclass may not declare additional instance variables.

c) A subclass may inherit methods or instance variables from its superclass but not both.

d) A subclass may inherit methods and instance variables from its superclass, and may also implement its own methods and declare its own instance variables.

Answer: d

Title: Which is true regarding subclasses?

Difficulty: Medium

Section Reference: 9.2 Implementing Subclasses

27. Which of the following is true regarding subclasses?

a) A subclass has access to private instance variables of its superclass.

b) A subclass does not have access to public instance variables of its superclass.

c) A subclass must specify the implicit parameter to use methods inherited from its superclass.

d) A subclass has no access to private instance variables of its superclass.

Answer: d

Title: Which is true regarding subclasses?

Difficulty: Medium

Section Reference: 9.2 Implementing Subclasses

28. Which of the following indicates that a class named Class1 is a subclass of a class named Class2?

a) public class Class1 extends Class2

b) public class Class1 implements Class2

c) public class Class2 extends Class1

d) public class Class2 implements Class1

Answer: a

Title: Which indicates that the Class1 class is a subclass of the Class2 class?

Difficulty: Easy

Section Reference: 9.2 Implementing Subclasses

29. Which of the following indicates that a class named ClassA class is a superclass of the ClassB class?

a) public class ClassB extends ClassA

b) public class ClassB implements ClassA

c) public class ClassA extends ClassB

d) public class ClassA implements ClassB

Answer: a

Title: Which indicates that the ClassA class is a superclass of the ClassB class?

Difficulty: Easy

Section Reference: 9.2 Implementing Subclasses

30. What must a subclass do to modify a private superclass instance variable?

a) The subclass must simply use the name of the superclass instance variable.

b) The subclass must declare its own instance variable with the same name as the superclass instance variable.

c) The subclass must use a public method of the superclass (if it exists) to update the superclass's private instance variable.

d) The subclass must have its own public method to update the superclass's private instance variable.

Answer: c

Title: How to modify a private superclass instance variable

Difficulty: Hard

Section Reference: 9.2 Implementing Subclasses

31. Which of the following indicates that the Motorcycle class is a subclass of the Vehicle class?

a) public class Motorcycle extends Vehicle

b) public class Motorcycle implements Vehicle

c) public class Vehicle extends Motorcycle

d) public class Vehicle implements Motorcycle

Answer: a

Title: Which indicates that the Motorcycle class is a subclass of the Vehicle class?

Difficulty: Easy

Section Reference: 9.2 Implementing Subclasses

32. Consider the following code snippet:

public class Vehicle

{

private String manufacturer;

. . .

public void setVehicleClass(double numberAxles)

{

. . .

}

}

If a Motorcycle class is created as a subclass of the Vehicle class, which of the following statements is correct?

a) A Motorcycle object inherits and can directly use both the instance variable manufacturer and the method setVehicleClass.

b) A Motorcycle object inherits and can directly use the instance variable manufacturer but not the method setVehicleClass.

c) A Motorcycle object inherits but cannot directly use either the instance variable manufacturer or the method setVehicleClass.

d) A Motorcycle object inherits and can directly use the method setVehicleClass but cannot directly use the instance variable manufacturer.

Answer: d

Title: Which statement about inheritance is correct?

Difficulty: Hard

Section Reference: 9.2 Implementing Subclasses

33. Consider the following code snippet:

public class Vessel

{

private String manufacturer;

. . .

public void setVesselClass(double engineRPM)

{

. . .

}

}

If a Speedboat class is created as a subclass of the Vessel class, which of the following statements is correct?

a) A Speedboat object inherits and can directly use both the instance variable manufacturer and the method setVesselClass.

b) A Speedboat object inherits and can directly use the instance variable manufacturer but not the method setVesselClass.

c) A Speedboat object inherits but cannot directly use either the instance variable manufacturer or the method setVesselClass.

d) A Speedboat object inherits and can directly use the method setVesselClass but cannot directly use the instance variable manufacturer.

Answer: d

Title: Which statement about inheritance is correct?

Difficulty: Hard

Section Reference: 9.2 Implementing Subclasses

34. Consider the following class hierarchy:

public class Vehicle

{

private String type;

public Vehicle(String type)

{

this.type = type;

}

public String displayInfo()

{

return type;

}

}

public class LandVehicle extends Vehicle

{

public LandVehicle(String type)

{

super(type);

}

}

public class Auto extends LandVehicle

{

public Auto(String type)

{

super(type);

}

}

You have written a program to use these classes, as shown in the following code snippet:

public class VehicleTester

{

public static void main(String[] args)

{

Auto myAuto = new Auto("sedan");

System.out.println("MyAuto type = " + \_\_\_\_\_\_);

}

}

Complete the code in this program snippet to correctly display the auto's type.

a) myAuto.displayInfo()

b) myAuto.super.displayInfo()

c) myAuto.super.super.displayInfo()

d) This cannot be done unless the Auto class overrides the displayInfo method.

Answer: a

Title: Complete the code to use a subclass

Difficulty: Medium

Section Reference: 9.2 Implementing Subclasses

35. Consider the following class hierarchy:

public class Vehicle

{

private String type;

public Vehicle(String type)

{

this.type = type;

}

public String displayInfo()

{

return type;

}

}

public class LandVehicle extends Vehicle

{

public LandVehicle(String type)

{

super(type);

}

}

public class Auto extends LandVehicle

{

public Auto(String type)

{

\_\_\_\_\_\_\_\_\_;

}

}

Complete the code in the Auto class constructor to store the type data.

a) super(type);

b) super(super(type));

c) super.super(type);

d) This cannot be done unless the Auto declares an instance variable named type.

Answer: a

Title: Complete the code to use a subclass

Difficulty: Medium

Section Reference: 9.2 Implementing Subclasses

36. Consider the following class hierarchy:

public class Vehicle

{

private String type;

public Vehicle(String type)

{

this.type = type;

}

public String displayInfo()

{

return type;

}

}

public class LandVehicle extends Vehicle

{

public LandVehicle(String type)

{

. . .

}

}

public class Auto extends LandVehicle

{

public Auto(String type)

{

. . .

}

public String displayAutoType()

{

return \_\_\_\_\_;

}

}

Complete the code in the Auto class method named displayAutoType to return the type data.

a) super(type);

b) super.type;

c) super.super.type;

d) super.displayInfo()

Answer: d

Title: Complete the code to use a subclass

Difficulty: Medium

Section Reference: 9.2 Implementing Subclasses

37. Which of the following statements about superclasses and subclasses is true?

a) A superclass is larger than its subclass.

b) A superclass inherits from a subclass.

c) A superclass extends a subclass.

d) A subclass extends a superclass.

Answer: d

Title: Which statement about superclasses and subclasses is true?

Difficulty: Medium

Section Reference: Common Error 9.2 Confusing Super- and Subclasses

38. Consider the following code snippet:

public class Vehicle

{

. . .

public void setVehicleAtrributes()

{

. . .

}

}

public class Auto extends Vehicle

{

. . .

public void setVehicleAtrributes()

{

. . .

}

}

Which of the following statements is correct?

a) The subclass is shadowing a superclass method.

b) The subclass is overloading a superclass method.

c) The subclass is overriding a superclass method.

d) This code will not compile.

Answer: c

Title: Which statement is correct about this code?

Difficulty: Medium

Section Reference: 9.2 Implementing Subclasses

39. Consider the following code snippet:

public class Vessel

{

. . .

public void setVesselAtrributes()

{

. . .

}

}

public class Speedboat extends Vessel

{

. . .

public void setVesselAtrributes()

{

. . .

}

}

Which of the following statements is correct?

a) The subclass is shadowing a superclass method.

b) The subclass is overloading a superclass method.

c) The subclass is overriding a superclass method.

d) This code will not compile.

Answer: c

Title: Which statement is correct about this code?

Difficulty: Medium

Section Reference: 9.2 Implementing Subclasses

40. Consider the following code snippet:

public void deposit(double amount)

{

transactionCount ++;

super.deposit(amount);

}

Which of the following statements is true?

a) This method will call itself.

b) This method calls a public method in its subclass.

c) This method calls a private method in its superclass

d) This method calls a public method in its superclass.

Answer: d

Title: Which statement about using the super keyword is true?

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

41. Which reserved word must be used to call a method of a superclass?

a) this

b) my

c) parent

d) super

Answer: d

Title: Which reserved word is used to call a method of a superclass?

Difficulty: Easy

Section Reference: 9.3 Overriding Methods

42. If a subclass defines the same method name and the same parameter types for a method that appears in its superclass, \_\_\_\_.

a) the subclass method overloads the superclass method.

b) the subclass method overrides the superclass method.

c) the subclass has implemented its superclass's method.

d) a compiler error will occur.

Answer: b

Title: When a subclass uses the same method name and the same parameter types

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

43. Consider the following code snippet:

public class Vehicle

{

. . .

public void setVehicleClass(double numberAxles)

{

. . .

}

}

public class Motorcycle extends Vehicle

{

. . .

public void setVehicleClass(double numberAxles)

{

. . .

}

}

Which of the following statements is correct? (I changed the wording below, because methods override methods. . .classes don't override methods)

a) The Motorcycle class's setVehicleClass method overrides the Vehicle class's setVehicleClass method.

b) The Vehicle class's setVehicleClass method overrides the Motorcycle class's setVehicleClass method.

c) The Motorcycle class's setVehicleClass method overloads the Vehicle class's setVehicleClass method.

d) The Vehicle class's setVehicleClass method overloads the Motorcycle class's setVehicleClass method.

Answer: a

Title: Which statement is correct about this code?

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

44. Consider the following code snippet:

public class Vessel

{

. . .

public void setVesselClass(double numberAxles)

{

. . .

}

}

public class Speedboat extends Vessel

{

. . .

public void setVesselClass(double numberAxles)

{

. . .

}

}

Which of the following statements is correct?

a) The Speedboat class's setVesselClass method overrides the Vessel class's setVesselClass method.

b) The Vessel class's setVesselClass method overrides the Speedboat class's setVesselClass method.

c) The Speedboat class's setVesselClass method overloads the Vessel class's setVesselClass method.

d) The Vessel class's setVesselClass method overloads the Speedboat class's setVesselClass method.

Answer: a

Title: Which statement is correct about this code?

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

45. Consider the following code snippet:

public class Vehicle

{

. . .

public void setVehicleClass(double numberAxles)

{

. . .

}

}

public class Motorcycle extends Vehicle

{

. . .

public void setModelName(String model)

{

. . .

}

public void setVehicleClass(double numberAxles)

{

. . .

}

}

Which of the following statements is NOT correct?

a) The Motorcycle class can call the setVehicleClass method of the Vehicle class.

b) The Vehicle class can call the setModelName method.

c) The Motorcycle class's SetVehicleClass method overrides the Vehicle class's setVehicleClass method.

d) The Motorcycle class can call the setVehicleClass method of the Motorcycle class.

Answer: b

Title: Which statement about inheritance is NOT correct?

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

46. Consider the following code snippet:

public class Employee

{

. . .

public void setDepartment(String deptName)

{

. . .

}

}

public class Programmer extends Employee

{

. . .

public void setProjectName(String projName)

{

. . .

}

public void setDepartment(String deptName)

{

. . .

}

}

Which of the following statements is NOT correct?

a) The Programmer class can call the setDepartment method of the Employee class.

b) The Employee class can call the setProjectName method. c) The Programmer class's setDepartment method overrides the Employee class's setDepartment method.

d) The Programmer class can call the setDepartment method of the Programmer class.

Answer: b

Title: Which statement about inheritance is NOT correct?

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

47. To override a superclass method in a subclass, the subclass method \_\_\_\_.

a) Must use a different method name.

b) Must use the same method name and the same parameter types.

c) Must use a different method name and the same parameter types.

d) Must use a different method name and different parameter types.

Answer: b

Title: How to override a superclass method in a subclass

Difficulty: Medium

Learning Objective : LO 9.3 Implement methods that override methods from a superclass.

Section Reference: Common Error 9.3 Accidental Overloading

48. Consider the following code snippet:

public class Vehicle

{

. . .

public void setVehicleClass(double numberAxles)

{

. . .

}

}

public class Auto extends Vehicle

{

. . .

public void setVehicleClass(int numberAxles)

{

. . .

}

}

Which of the following statements is correct?

a) The Auto class overrides the setVehicleClass method.

b) The Vehicle class overrides the setVehicleClass method.

c) The Auto class overloads the setVehicleClass method.

d) The Vehicle class overloads the setVehicleClass method.

Answer: c

Title: How to modify a superclass method in a subclass

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

Section Reference 2: Common Error 9.3 Accidental Overloading

49. Consider the following code snippet:

public class Employee

{

. . .

public void setEmployeeDept(String deptNum)

{

. . .

}

}

public class Programmer extends Employee

{

. . .

public void setEmployeeDept(int deptNum)

{

. . .

}

}

Which of the following statements is correct?

a) The Programmer class overrides the setEmployeeDept method.

b) The Employee class overrides the setEmployeeDept method.

c) The Programmer class overloads the setEmployeeDept method.

d) The Employee class overloads the setEmployeeDept method.

Answer: c

Title: How to modify a superclass method in a subclass

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

Section Reference 2: Common Error 9.3 Accidental Overloading

50. If a subclass contains a method with the same name as a method in its superclass, but with different parameter types, the subclass method is said to \_\_\_\_ the method of the superclass.

a) implement

b) inherit

c) override

d) overload

Answer: d

Title: When a subclass has same method name as its superclass

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

Section Reference 2: Common Error 9.3 Accidental Overloading

51. If a subclass uses the same method name but different parameter types for a method that appears in its superclass, \_\_\_\_.

a) the subclass method has overloaded its superclass's method.

b) the subclass method has overridden its superclass's method.

c) the subclass has implemented its superclass's method.

d) a compiler error will occur.

Answer: a

Title: When a subclass uses the same method name but different parameter types

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

Section Reference 2: Common Error 9.3 Accidental Overloading

52. Consider the following code snippet that appears in a subclass:

public void deposit(double amount)

{

transactionCount ++;

deposit(amount);

}

Which of the following statements is true?

a) This method will call itself.

b) This method calls a public method in its subclass.

c) This method calls a private method in its superclass

d) This method calls a public method in its superclass.

Answer: a

Title: Which statement is correct about this code?

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

Section Reference 2: Common Error 9.4 Forgetting to use super

53. Consider the following code snippet:

public class Auto extends Vehicle

{

. . .

public Auto(int numberAxles)

{

super(numberAxles);

}

}

What does this code do?

a) It invokes the constructor of the Vehicle class from within the constructor of the Auto class.

b) It invokes the constructor of the Auto class from within the constructor of the Vehicle class.

c) It invokes a private method of the Vehicle class from within a method of the Auto class.

d) This code will not compile.

Answer: a

Title: What does the keyword super do?

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

Section Reference 2: Special Topic 9.1 Calling the Superclass Constructor

54. Consider the following code snippet:

public class Motorcycle extends Vehicle

{

. . .

public Motorcycle(int numberAxles)

{

super(numberAxles);

}

}

What does this code do?

a) It invokes the constructor of the Vehicle class from within the constructor of the Motorcycle class.

b) It invokes the constructor of the Motorcycle class from within the constructor of the Vehicle class.

c) It invokes a private method of the Vehicle class from within a method of the Motorcycle class.

d) This code will not compile.

Answer: a

Title: What does the keyword super do?

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

Section Reference 2: Special Topic 9.1 Calling the Superclass Constructor

55. Consider the following code snippet:

public class Motorcycle extends Vehicle

{

. . .

public Motorcycle(int numberAxles)

{

super(numberAxles); //line #1

}

}

If the line marked "//line #1" was missing, which of these statements would be correct?

a) The Motorcycle class constructor would invoke the constructor of the Vehicle class with no parameters.

b) The Vehicle class constructor would invoke the constructor of the Motorcycle class with no parameters.

c) The Motorcycle class constructor would invoke the constructor of the Vehicle class with a parameter value of 0.

d) This code would not compile.

Answer: a

Title: What does this code do?

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

Section Reference 2: Special Topic 9.1 Calling the Superclass Constructor

56. Consider the following code snippet:

public class Motorcycle extends Vehicle

{

. . .

public Motorcycle(int numberAxles)

{

super.numberAxles;

}

}

What does this code do?

a) It invokes the constructor of the Vehicle class from within the constructor of the Motorcycle class.

b) It invokes the constructor of the Motorcycle class from within the constructor of the Vehicle class.

c) It invokes a private method of the Vehicle class from within a method of the Motorcycle class.

d) This code will not compile.

Answer: d

Title: What does the keyword super do?

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

Section Reference 2: Special Topic 9.1 Calling the Superclass Constructor

57. Consider the following code snippet:

public class Motorcycle extends Vehicle

{

private String model;

. . .

public Motorcycle(int numberAxles, String modelName)

{

model = modelName;

super(numberAxles);

}

}

What does this code do?

a) It invokes the constructor of the Vehicle class from within the constructor of the Motorcycle class.

b) It invokes the constructor of the Motorcycle class from within the constructor of the Vehicle class.

c) It invokes a private method of the Vehicle class from within a method of the Motorcycle class.

d) This code will not compile.

Answer: d

Title: What does the keyword super do?

Difficulty: Medium

Section Reference: Special Topic 9.1 Calling the Superclass Constructor

58. Consider the following code snippet:

public class Motorcycle extends Vehicle

{

private String model;

. . .

public Motorcycle(int numberAxles, String modelName)

{

super(numberAxles);

model = modelName;

}

}

What does this code do?

a) It invokes the constructor of the Vehicle class from within the constructor of the Motorcycle class.

b) It invokes the constructor of the Motorcycle class from within the constructor of the Vehicle class.

c) It invokes a private method of the Vehicle class from within a method of the Motorcycle class.

d) This code will not compile.

Answer: a

Title: What does the keyword super do?

Difficulty: Medium

Section Reference: 9.3 Overriding Methods

Section Reference 2: Special Topic 9.1 Calling the Superclass Constructor

59. Consider the classes shown below:

public class Parent

{

public void doSomething() // method 1

{ /\* Implementation not shown \*/ }

}

public class Child extends Parent

{

public void doSomething(int n) // method 2

{ /\* Implementation not shown \*/ }

public void doSomething() // method 3

{ /\* Implementation not shown \*/ }

}

If the variable kid is defined below, which version of the doSomething method can be called on the variable kid?

Child kid = new Child();

a) Methods 1 and 2 only

b) Method 2 only

c) Methods 2 and 3 only

d) Methods 1, 2, and 3

Answer: b

Title: Which version of the doSomething method can be called on the variable kid?

Difficulty: Medium

Section Reference 1: 9.3 Overriding Methods

60. Consider the classes shown below:

public class Parent

{

public void doSomething() // method 1

{ /\* Implementation not shown \*/ }

}

public class Child extends Parent {

public void doSomething(int n) // method 2

{ /\* Implementation not shown \*/ }

public void doSomething() // method 3

{ /\* Implementation not shown \*/ }

}

If the variable kid is defined below, which version of the doSomething method can be called on the variable kid?

Parent kid = new Child();

a) Method 1 only

b) Methods 2 and 3 only

c) Methods 1 and 2 only

d) Methods 1, 2, and 3

Answer: a

Title: Which version of the doSomething method can be called on the variable kid?

Difficulty: Medium

Section Reference 1: 9.3 Overriding Methods

61. Consider the classes shown below:

public class Parent

{

public int getValue()

{

return 24;

}

public void display()

{

System.out.print(getValue() + " ");

}

}

public class Child extends Parent

{

public int getValue()

{

return -7;

}

}

Using the classes above, what is the output of the following lines of code?

Child kid = new Child();

Parent adult = new Parent();

kid.display();

adult.display();

a) 24 24

b) -7 -7

c) -7 24

d) 24 -7

Answer: c

Title: What is the output of the following lines of code?

Difficulty: Medium

Section Reference 1: 9.4 Polymorphism

62. Consider the classes shown below:

public class Parent

{

public int getValue()

{

return 24;

}

public void display()

{

System.out.print(getValue() + " ");

}

}

public class Child extends Parent

{

public int getValue()

{

return -7;

}

}

Using the classes above, what is the output of the following lines of code?

Parent kid = new Child();

Parent adult = new Parent();

kid.display();

adult.display();

a) -7 24

b) 24 24

c) -7 -7

d) 24 -7

Answer: a

Title: What is the output of the following lines of code?

Difficulty: Medium

Section Reference 1: 9.4 Polymorphism

63. Suppose the abstract class Message is defined below

public abstract class Message {

private String value;

public Message(String initial)

{

value = initial;

}

public String getMessage()

{

return value;

}

public abstract String translate();

}

A concrete subclass of Message, FrenchMessage, is defined. Which methods must FrenchMessage define?

a) translate() only

b) getMessage() only

c) The FrenchMessage constructor and translate() only

d) The FrenchMessage constructor, getMessage(), and translate()

Answer: a

Title: Which methods must FrenchMessage define?

Difficulty: Medium

Section Reference 1: 9.4 Polymorphism

64. Consider the Counter class below.

public class Counter

{

public int count = 0;

public int getCount()

{

return count;

}

public void increment()

{

count++;

}

}

Using the class above and the variable declared below, what is the value of num.toString()?

Counter num = new Counter();

a) a string with count's value

b) a string with num's type and hashcode

c) a string with count's type and hashcode

d) nothing since toString is not defined for Counter

Answer: b

Title: What is the value of num.toString()?

Difficulty: Medium

Section Reference 1: 9.5 Object: The Cosmic Superclass

65. Consider the Counter class below.

public class Counter

{

public int count = 0;

public int getCount()

{

return count;

}

public void increment()

{

count++;

}

}

Using the class above and the variables declared below, what is the value of num1.equals(num2)?

Counter num1 = new Counter();

Counter num2 = new Counter();

a) true

b) false

c) nothing since equals is not defined for Counter

Answer: b

Title: What is the value of num1.equals(num2)?

Difficulty: Medium

Section Reference 1: 9.5 Object: The Cosmic Superclass

66. Consider the Counter class below.

public class Counter

{

public int count = 0;

public int getCount()

{

return count;

}

public void increment()

{

count++;

}

}

Using the class above and the variables declared below, what is the value of num1.equals(num2)?

Counter num1 = new Counter();

Counter num2 = num1;

a) true

b) false

c) nothing since equals is not defined for Counter

Answer: a

Title: What is the value of num1.equals(num2)?

Difficulty: Medium

Section Reference 1: 9.5 Object: The Cosmic Superclass

67. Consider the following class hierarchy:

public class Vehicle

{

private String type;

public Vehicle(String type)

{

this.type = type;

}

public String getType()

{

return type;

}

}

public class LandVehicle extends Vehicle

{

public LandVehicle(String type)

{

. . .

}

}

public class Auto extends LandVehicle

{

public Auto(String type)

{

. . .

}

}

Which of the following code fragments is NOT valid in Java?

a) Vehicle myAuto = new Auto("sedan");

b) LandVehicle myAuto = new Auto("sedan");

c) Auto myAuto = new Auto("sedan");

d) LandVehicle myAuto = new Vehicle("sedan");

Answer: d

Title: Which code fragment is NOT valid?

Difficulty: Medium

Section Reference: 9.4 Polymorphism

68) Consider the following code snippet:

Vehicle aVehicle = new Auto();

aVehicle.moveForward(200);

If the Auto class inherits from the Vehicle class, and both classes have an implementation of the moveForward method with the same set of parameters and the same return type, which statement is correct?

a) The moveForward method of the Auto class will be executed.

b) The moveForward method of the Vehicle class will be executed.

c) You must specify in the code which class's moveForward method is to be used.

d) It is not possible to determine which class's method is called.

Answer: a

Title: Which method will be executed?

Difficulty: Medium

Section Reference: 9.4 Polymorphism

69) Consider the following code snippet:

Employee anEmployee = new Programmer();

anEmployee.increaseSalary(2500);

If the Programmer class inherits from the Employee class, and both classes have an implementation of the increaseSalary method with the same set of parameters and the same return type, which statement is correct?

a) The increaseSalary method of the Programmer class will be executed.

b) The increaseSalary method of the Employee class will be executed.

c) You must specify in the code which class's increaseSalary method is to be used.

d) It is not possible to determine which class's method is called.

Answer: a

Title: Which method will be executed?

Difficulty: Medium

Section Reference: 9.4 Polymorphism

70. Consider the following code snippet:

Vehicle aVehicle = new Auto();

aVehicle.moveForward(200);

Assume that the Auto class inherits from the Vehicle class, and both classes have an implementation of the moveForward method with the same set of parameters and the same return type. The process for determining which class's moveForward method to execute is called \_\_\_\_.

a) inheritance disambiguation.

b) inheritance hierarchy.

c) dynamic inheritance.

d) dynamic lookup.

Answer: d

Title: The process to determine which method is executed is called \_\_\_\_.

Difficulty: Medium

Section Reference: 9.4 Polymorphism

71. Consider the following code snippet:

Vehicle aVehicle = new Auto();

aVehicle.moveForward(200);

Assume that the Auto class inherits from the Vehicle class, and both classes have an implementation of the moveForward method with the same set of parameters and the same return type. Which class's moveForward method is to be executed is determined by \_\_\_\_.

a) the actual object type.

b) the variable's type.

c) the hierarchy of the classes.

d) it is not possible to determine which method is executed.

Answer: a

Title: The method to be executed is determined by \_\_\_\_.

Difficulty: Medium

Section Reference: 9.4 Polymorphism

72. Consider the following code snippet:

Employee anEmployee = new Programmer();

anEmployee.increaseSalary(2500);

Assume that the Programmer class inherits from the Employee class, and both classes have an implementation of the increaseSalary method with the same set of parameters and the same return type. Which class's increaseSalary method is to be executed is determined by \_\_\_\_.

a) the hierarchy of the classes.

b) the variable's type.

c) the actual object type.

d) it is not possible to determine which method is executed.

Answer: c

Title: The method to be executed is determined by \_\_\_\_.

Difficulty: Medium

Section Reference: 9.4 Polymorphism

73. Which of the following statements about abstract methods is true?

a) An abstract method has a name, parameters, and a return type, but no code in the body of the method.

b) An abstract method has parameters, a return type, and code in its body, but has no defined name.

c) An abstract method has a name, a return type, and code in its body, but has no parameters.

d) An abstract method has only a name and a return type, but no parameters or code in its body.

Answer: a

Title: Which statement about abstract methods is true?

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.3 Abstract Classes

74. Which of the following statements about classes is true?

a) You can create an object from a concrete class, but not from an abstract class.

b) You can create an object from an abstract class, but not from a concrete class.

c) You cannot have an object reference whose type is an abstract class.

d) You cannot create subclasses from abstract classes.

Answer: a

Title: Which statement about classes is true?

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.3 Abstract Classes

75. If a class has an abstract method, which of the following statements is NOT true?

a) You can construct an object from this class.

b) You can have an object reference whose type is this class.

c) You can inherit from this class.

d) All non-abstract subclasses of this class must implement this method.

Answer: a

Title: Which statement about classes with abstract methods is NOT true?

Difficulty: Hard

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.3 Abstract Classes

76. Consider the following code snippet:

public abstract class Machine

{

public abstract void setRPMs();

. . .

}

You wish to create a concrete subclass named PolisherMachine. Which of the following is the correct way to declare this subclass?

a)

public class PolisherMachine implements Machine

{

public void setRPMs() { . . . }

}

b)

public class PolisherMachine extends Machine

{

void setRPMs() { . . . }

}

c)

public class PolisherMachine implements Machine

{

void setRPMs() { . . . }

}

d)

public class PolisherMachine extends Machine

{

public void setRPMs() { . . . }

}

Answer: d

Title: Which code creates a subclass from this class?

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.3 Abstract Classes

77. Consider the following code snippet:

public abstract class Employee

{

public abstract void setSalary();

. . .

}

You wish to create a concrete subclass named Programmer. Which of the following is the correct way to declare this subclass?

a)

public class Programmer implements Employee

{

public void setSalary() { . . . }

}

b)

public class Programmer extends Employee

{

void setSalary() { . . . }

}

c)

public class Programmer implements Employee

{

void setSalary() { . . . }

}

d)

public class Programmer extends Employee

{

public void setSalary() { . . . }

}

Answer: d

Title: Which code creates a subclass from this class?

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.3 Abstract Classes

78. A class from which you cannot create objects is called a/an \_\_\_\_.

a) Abstract class.

b) Concrete class.

c) Non-inheritable class.

d) Superclass.

Answer: a

Title: Which statement about classes is true?

Difficulty: Easy

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.3 Abstract Classes

79. A class that cannot be instantiated is called a/an \_\_\_\_.

a) Abstract class.

b) Anonymous class.

c) Concrete class.

d) Non-inheritable class.

Answer: a

Title: Which statement about classes is true?

Difficulty: Easy

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.3 Abstract Classes

80. Which of the following statements about classes is true?

a) You can create an object from a class declared with the keyword final.

b) You can override methods in a class declared with the keyword final.

c) You can extend a class declared with the keyword final.

d) You can create subclasses from a class declared with the keyword final.

Answer: a

Title: Which statement about classes is NOT true?

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.4 Final Methods and Classes

81. The \_\_\_\_ reserved word in a class definition ensures that subclasses cannot be created from this class.

a) abstract

b) anonymous

c) final

d) static

Answer: c

Title: Prohibiting the creation of subclasses from a class

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.4 Final Methods and Classes

82. The \_\_\_\_ reserved word in a method definition ensures that subclasses cannot override this method.

a) abstract

b) anonymous

c) final

d) static

Ans: c

Title: How to ensure a method cannot be overridden

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.4 Final Methods and Classes

83. Which of the following is true regarding inheritance?

a) When creating a subclass, all methods of the superclass must be overridden.

b) When creating a subclass, no methods of a superclass can be overridden.

c) A superclass can force a programmer to override a method in any subclass created from it.

d) A superclass cannot prevent a programmer from overriding a method in any subclass created from it.

Answer: c

Title: Which statement regarding inheritance is true?

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.4 Final Methods and Classes

84. When declared as protected, data in an object can be accessed by \_\_\_\_.

a) Only by that class's methods and by all of its subclasses

b) Only by that class's methods, by all of its subclasses, and by methods in classes within the same package.

c) Only by that class's methods.

d) By any class.

Answer: b

Title: Protected data in an object can be accessed by \_\_\_\_.

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.5 Protected Access

85. Consider the following code snippet:

public class Vehicle

{

protected int numberAxles;

. . .

}

Data in the numberAxles variable can be accessed by \_\_\_\_.

a) Only by the Vehicle class's methods and by all of its subclasses

b) Only by the Vehicle class's methods, by all of its subclasses, and by methods in classes within the same package.

c) Only by the Vehicle class's methods.

d) By any class.

Answer: b

Title: Protected data in an object can be accessed by \_\_\_\_.

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.5 Protected Access

86. To ensure that an instance variable can only be accessed by the class that declared it, the variable should be declared as \_\_\_\_.

a) public

b) private

c) protected

d) final

Answer: b

Title: Controlling access to an instance variable

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.5 Protected Access

87. With a few exceptions, instance variables of classes should always have \_\_\_ access.

a) final

b) private

c) public

d) protected

Answer: b

Title: Instance variables of classes should always have \_\_\_ access.

Difficulty: Medium

Section Reference: 9.4 Polymorphism

Section Reference 2: Special Topic 9.5 Protected Access

88. Consider the following code snippet:

Vehicle aVehicle = new Auto(4,"gasoline");

String s = aVehicle.toString();

Assume that the Auto class inherits from the Vehicle class, and neither class has an implementation of the toString() method. Which of the following statements is correct?

a) The toString() method of the Object class will be used when this code is executed. b) The toString() method of the String class will be used when this code is executed.

c) This code will not compile because there is no toString() method in the Vehicle class.

d) This code will not compile because there is no toString() method in the Auto class.

Answer: a

Title: Which statement about the toString method is correct?

Difficulty: Medium

Section Reference: 9.5 Object: The Cosmic Superclass

89. Consider the following code snippet:

Employee anEmployee = new Programmer();

String emp = anEmployee.toString();

Assume that the Programmer class inherits from the Employee class, and neither class has an implementation of the toString() method. Which of the following statements is correct?

a) The toString() method of the Object class will be used when this code is executed.

b) The toString() method of the String class will be used when this code is executed.

c) This code will not compile because there is no toString() method in the Employee class.

d) This code will not compile because there is no toString() method in the Programmer class.

Answer: a

Title: Which statement about the toString method is correct?

Difficulty: Medium

Section Reference: 9.5 Object: The Cosmic Superclass

90. Consider the following code snippet:

int numAxles = 4;

String s = "Number of axles is " + numAxles;

Which of the following statements is correct?

a) The toString() method of the Object class is being used to set the value of s.

b) The toString() method of the Integer class is being used to set the value of s.

c) No toString() method is being used to set the value of s.

d) This code will not compile.

Answer: c

Title: Which statement about this code is correct?

Difficulty: Medium

Section Reference: 9.5 Object: The Cosmic Superclass

Section Reference 2: 9.5.1 Overriding the toString method

91. Consider the following code snippet:

int vacationDays = 10;

String output = "Number of earned vacation days is " + vacationDays;

Which of the following statements is correct?

a) The toString() method of the Object class is being used to set the value of output.

b) The toString() method of the Integer class is being used to set the value of output.

c) No toString() method is being used to set the value of output.

d) This code will not compile.

Answer: c

Title: Which statement about this code is correct?

Difficulty: Medium

Section Reference: 9.5 Object: The Cosmic Superclass

Section Reference 2: 9.5.1 Overriding the toString method

92. Consider the following code snippet:

double salary = 45000.00;

String sal = "Current salary is " + salary;

Which of the following statements is correct?

a) The toString() method of the Object class is being used to set the value of salary.

b) The toString() method of the Double class is being used to set the value of salary.

c) No toString() method is being used to set the value of salary.

d) This code will not compile.

Answer: c

Title: Which statement about this code is correct?

Difficulty: Medium

Section Reference: 9.5 Object: The Cosmic Superclass

Section Reference 2: 9.5.1 Overriding the toString method

93. Consider the following code snippet:

Auto consumerAuto = new Auto(4, "gasoline");

String s = consumerAuto.toString();

Assume that the Auto class has not implemented its own toString() method. What value will s contain when this code is executed? a) s will contain the values of the instance variables in consumerAuto.

b) s will contain only the class name of the consumerAuto object.

c) s will contain the class name of the consumerAuto object followed by a hash code.

d) This code will not compile.

Answer: c

Title: What value will this variable have?

Difficulty: Medium

Section Reference: 9.5 Object: The Cosmic Superclass

Section Reference 2: 9.5.1 Overriding the toString method

94. Consider the following code snippet:

Employee programmer = new Employee(10254, "exempt");

String s = programmer.toString();

Assume that the Employee class has not implemented its own toString() method. What value will s contain when this code is executed?

a) s will contain the values of the instance variables in programmer.

b) s will contain only the class name of the programmer object.

c) s will contain the class name of the programmer object followed by a hash code.

d) This code will not compile.

Answer: c

Title: What value will this variable have?

Difficulty: Medium

Section Reference: 9.5 Object: The Cosmic Superclass

Section Reference 2: 9.5.1 Overriding the toString method

95. Which of the following statements about comparing objects is correct?

a) The equals method is used to compare whether two references are to the same object.

b) The equals method is used to compare whether two objects have the same contents.

c) The == operator is used to compare whether two objects have the same contents.

d) The equals method and the == operator perform the same actions.

Answer: b

Title: Which statement about comparing objects is correct?

Difficulty: Medium

Section Reference: 9.5 Object: The Cosmic Superclass

Section Reference 2: 9.5.2 The equals method

96. Consider the following code snippet, which is meant to override the equals() method of the Object class:

public class Coin

{

. . .

public boolean equals(Coin otherCoin)

{

. . .

}

. . .

}

What is wrong with this code?

a) A class cannot override the equals() method of the Object class.

b) The equals() method must be declared as private.

c) A class cannot change the parameters of a superclass method when overriding it.

d) There is nothing wrong with this code.

Answer: c

Title: What is wrong with this code?

Difficulty: Medium

Section Reference: 9.5 Object: The Cosmic Superclass

Section Reference 2: 9.5.2 The equals method

97. Consider the following code snippet:

public class Coin

{

private String name;

. . .

public boolean equals(Object otherCoin)

{

return name.equals(otherCoin.name);

}

. . .

}

What is wrong with this code?

a) The return statement should use the == operator instead of the equals method.

b) The parameter in the equals method should be declared as Coin otherCoin.

c) otherCoin must be cast as a Coin object before using the equals method.

d) There is nothing wrong with this code.

Answer: c

Title: What is wrong with this code that uses the equals method?

Difficulty: Hard

Section Reference: 9.5 Object: The Cosmic Superclass

Section Reference 2: 9.5.2 The equals method

98. Consider the following code snippet:

public class Score

{

private String name;

. . .

public boolean equals(Object otherScore)

{

return name.equals(otherScore.name);

}

. . .

}

What is wrong with this code?

a) The return statement should use the == operator instead of the equals method.

b) The parameter in the equals method should be declared as Score otherScore.

c) otherScore must be cast as a Score object before using the equals method.

d) There is nothing wrong with this code.

Answer: c

Title: What is wrong with this code that uses the equals method?

Difficulty: Hard

Section Reference: 9.5 Object: The Cosmic Superclass

Section Reference 2: 9.5.2 The equals method

99. Consider the following code snippet of a function object

public interface Measurer

{

double measure(\_\_\_\_\_\_ anObject);

}

Complete this code to allow the interface to handle all classes?

a) Class

b) Object

c) Any

d) Void

Answer: b

Title: Which can be used when interfaces are unavailable?

Difficulty: Medium

Section Reference: 9.6 Interface Types

Section Reference 2: Special Topic 9.9 Function Objects

100. Consider the following code snippet:

if(anObject instanceof Auto)

{

Auto anAuto = (Auto) anObject;

. . .

}

What does this code do?

a) This code tests whether anObject was created from a superclass of Auto.

b) This code creates a subclass type object from a superclass type object.

c) This class safely converts an object of any type to an object of type Auto.

d) This code safely converts an object of type Auto or a subclass of Auto to an object of type Auto.

Answer: d

Title: What does this code do?

Difficulty: Medium

Section Reference: 9.5 Object: The Cosmic Superclass

Section Reference 2: 9.5.3 The instanceOf operator

101. To test whether an object belongs to a particular type, use \_\_\_.

a) the this reserved word.

b) the subclassOf reserved word.

c) the instanceof operator.

d) the equals method.

Answer: c

Title: To test whether an object belongs to a particular type, use \_\_\_.

Difficulty: Medium

Section Reference: 9.5 Object: The Cosmic Superclass

Section Reference 2: 9.5.3 The instanceOf operator