HOMEWORK 7 JAVA INHERITANCE

Qn	Ans	Explanation
1	С	Two lines of the program contain compilation errors. The name variable in the Cinema Class is
		private so it cannot be accessible from Movie class. Movie Class is defined as a constructor and it
		is missing an explicit super() statement so Cinema Class does not include a no argument
		constructor.
3	D	Public modifier can be applied to an abstract interface method.
3	С	The code does not compile because Song class contains two methods with the same method
1	Α	signature. There is a conflict in the code.
5	A	Inheritance allows objects to access commonly used attributes and methods. Canine cannot be returned as an instance of Class because it does not inherit Class.
6	B	An interface allows easy integration once the other team's code is complete.
6 7	В	Output is "Driving electric car". The object created is an ElectricCar in the main() method, even if
,		it is assigned to a Car reference. Due to polymorphism, the method from the ElectricCar will be
		invoked so output will be "Driving electric car".
8	D	Java allows multiple inheritance using interfaces.
9	С	There are three problems with this method override. First problem is that watch() method in the
		Television class is marked final. Second problem is that the return types void and Object are not
		covariant. Last problem is that access modifier in the child class must be the same or broader than
10	С	in the parent class. A checked exception thrown by the method in the parent class must be thrown by the method in the
10		child class. Option C is incorrect.
11	С	The code does not compile because the process() method is declared final in the Computer class.
12	A	The output is "2". Due to polymorphism, the overridden version of the method in HighSchool is
		used, regardless of the reference type, and 2 is printed.
13	В	Static modifier can be applied to an interface method.
14	C	Having one class implement two interfaces that both define the same default method
		signature leads to a compiler error, unless the class overrides the default method. In this
		case, the Sprint class does override the walk() method correctly, therefore the code
		compiles without issue, and Option C is correct.
15	В	Option B is not true. An interface can implement another interface.
16	D	The code does not compile because super.height is not visible in the Rocket class, making
		Option D the correct answer. Even though the Rocket class defines a height value, the
		super keyword looks for an inherited version. Since there are none, the code does not
		compile. Note that super.getWeight() returns 3 from the variable in the parent class, as
		polymorphism and overriding does not apply to instance variables.
17	D	Excluding default and static methods, an abstract class can contain both abstract and concrete
		methods, while an interface contains only abstract methods.
18	C	The code does not compile, so Option D is incorrect. The IsoscelesRightTriangle class is
		abstract; therefore, it cannot be instantiated on line g3. Only concrete classes can be
		instantiated, so the code does not compile, and Option C is the correct answer. The rest of
		the lines of code compile without issue. A concrete class can extend an abstract class, and
		an abstract class can extend a concrete class. Also, note that the override of
		getDescription() has a widening access modifier, which is fine per the rules of overriding
		methods.
19	D	The play() method is overridden in Saxophone for both Horn and Woodwind, so the
		return type must be covariant with both. Unfortunately, the inherited methods must also
		be compatible with each other. Since Integer is not a subclass of Short, and vice versa,
		there is no subclass that can be used to fill in the blank that would allow the code to
		compile. In other words, the Saxophone class cannot compile regardless of its
	<u> </u>	implementation of play(), making Option D the correct answer.
20	С	A class implements an interface, while a class extends an abstract class.
21	A	The code compiles and runs without issue, making Options C and D incorrect. Although
		super.material and this.material are poor choices in accessing static variables, they are

returned, making Option A the correct answer. Also, note that the constructor Book(String) is not used in the Encyclopedia class. 1		1	
Book(String) is not used in the Encyclopedia class. 22 B The data type of unknownBunny must be Bunny or a subclass of Bunny. 23 D Protected modifiers can be applied to an abstract method. 24 D The declaration of Sphere compiles without issue, so Option C is incorrect. The Mars class declaration is invalid because Mars cannot extend Sphere, an interface, nor can Mars implement Planet, a class. In other words, they are reversed. Since the code does not compile, Option D is the correct answer. Note that if Sphere and Planet were swapped in the Mars class definition, the code would compile and the output would be Mars, making Option A the correct answer. Note that if Sphere and Planet were swapped in the Mars class definition, the code would compile and the output would be Mars, making Option A the correct answer. 25 B A Areference to a class can be assigned to a superclass reference without an explicit cast. 26 B Abstract is not implicitly applied to all interface variables. 27 C The output is "1245". The class is loaded first, with the static initialization block called and I is outputted first. When the BlueCar is created in the main() method, the superclass initialization blappens first. The instance initialization blocks are executed before the constructor, so 32 is outputted next. Finally, the class is loaded with the instance initialization blocks again being called before the constructor, outputing 45. 28 C Overloaded and overridden methods always have the same method name. 29 A The output is "5". At muttime, the object is passed around and, due to polymorphism, can be read using any of those references since the underlying object is a SoccerBall. In other words, casting it to a different reference variable does not modify the object or cause it to lose its underlying SoccerBall information. 30 C A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable with the same signature as a stutic method in a parent class is referred t			permitted. Since super is used to access the variable in getMaterial(), the value papyrus is
B			
D			
24			
class declaration is invalid because Mars cannot extend Sphere, an interface, nor can Mars implement Planet, a class. In other words, they are reversed. Since the code does not compile, Option D is the correct answer. Note that if Sphere and Planet were swapped in the Mars class definition, the code would compile and the output would be Mars, making Option A the correct answer. 25 B A reference to a class can be assigned to a superclass reference without an explicit cast. 26 B Abstract is not implicitly applied to all interface variables. 27 C The output is "i3245". The class is loaded first, with the static initialization block called and 1 is outputted first. When the BlueCar is created in the main() method, the superclass initialization happens first. The instance initialization blocks are exceuted before the constructor, outputting 45. 28 C Overloaded and overridden methods always have the same method name. 29 A The output is "5". At runtime, the object is passed around and, due to polymorphism, can be read using any of those references since the underlying object is a SoccerBall. In other words, casting it to a different reference variable does not modify the object or cause it to lose its underlying SoccerBall information. 30 C A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a method. 31 B The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. 32 C The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. 33 B A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. 34 C A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstr			
implement Planet, a class. In other words, they are reversed. Since the code does not compile, Option D is the correct answer. Note that if Sphere and Planet were swapped in the Mars class definition, the code would compile and the output would be Mars, making Option A the correct answer. A reference to a class can be assigned to a superclass reference without an explicit cast. A bastract is not implicitly applied to all interface variables. The output is "13245". The class is loaded first, with the static initialization block called and 1 is outputted first. When the BlueCar is created in the main() method, the superclass initialization happens first. The instance imitialization blocks are executed before the constructor, so 23 is outputted next. Finally, the class is loaded with the instance initialization blocks again being called before the constructor, outputting 45. Overloaded and overridded with the instance initialization blocks again being called before the constructor, outputting 45. Overloaded and overridded methods always have the same method name. A The output is "5". At runtime, the object is passed around and, due to polymorphism, can be read using any of those references since the underlying object is a SoccerBall. In other words, casting it to a different reference variable does not modify the object or cause it to lose its underlying SoccerBall information. A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a method. The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A class that by a sassigned to a superclass reference variable automatically but requires an explicit cast when a	24	D	The declaration of Sphere compiles without issue, so Option C is incorrect. The Mars
compile, Option D is the correct answer. Note that if Sphere and Planet were swapped in the Mars class definition, the code would compile and the output would be Mars, making Option A the correct answer. B A reference to a class can be assigned to a superclass reference without an explicit cast. A bastract is not implicitly applied to all interface variables. The output is "13245". The class is loaded first, with the static initialization block called and 1 is outputted first. When the BlueCar is created in the main() method, the superclass initialization happens first. The instance initialization blocks are executed before the constructor, so 32 is outputted next. Finally, the class is loaded with the instance initialization blocks again being called before the constructor, outputing 45. B C Overloaded and overridden methods always have the same method name. The output is "5". At runtime, the object is passed around and, due to polymorphism, can be read using any of those references since the underlying object is a SoccerBall. In other words, casting it to a different reference variable does not modify the object or cause it to lose its underlying SoccerBall information. C A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a warriable with the same signature of the code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable automatically but requires an explicit cast when assigned to a superclas			class declaration is invalid because Mars cannot extend Sphere, an interface, nor can Mars
the Mars class definition, the code would compile and the output would be Mars, making Option A the correct answer. A reference to a class can be assigned to a superclass reference without an explicit cast. A reference to a class can be assigned to a superclass reference without an explicit cast. The output is "13245". The class is loaded first, with the static initialization block called and 1 is outputted first. When the BlueCar is created in the main() method, the superclass initialization happens first. The instance initialization blocks are executed before the constructor, so 32 is outputted next. Finally, the class is loaded with the instance initialization blocks again being called before the constructor, outputting 45. Overloaded and overridden methods always have the same method name. Provided an overridden methods always have the same method name. A The output is "5". At runtime, the object is passed around and, due to polymorphism, can be read using any of those references since the underlying object is a SoccerBall. In other words, casting it to a different reference variable does not modify the object or cause it to lose its underlying SoccerBall information. A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a method. The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A class may be assigned to a subclass reference variable. A class may be assigned to a subclass reference variable. A class may be assigned to a subclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A class may be assigned to a subclass reference variable automatically but			implement Planet, a class. In other words, they are reversed. Since the code does not
Option A the correct answer.			compile, Option D is the correct answer. Note that if Sphere and Planet were swapped in
A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method in parent class is referred to as hiding a wariable, while a class that defines a static method is not marked abstract. The code does not compile because the overrided automatically but requires an explicit cast when assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable automatically but requires an explicit cast when assigned to a subclass of hold for the content of fly(int speed) in the Bird class also does not compile because the class includes another class. A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method in a parent class is referred to as hiding a method. The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A class may be assigned to a subclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. A class may be assigned to a subclass had in the parent does with the same signature and parent does an abstract method. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because the class inherits two default methods with the same signature and no overridden ve			the Mars class definition, the code would compile and the output would be Mars, making
Abstract is not implicitly applied to all interface variables.			Option A the correct answer.
C	25	В	A reference to a class can be assigned to a superclass reference without an explicit cast.
outputted first. When the BlueCar is created in the main() method, the superclass initialization happens first. The instance initialization blocks are executed before the constructor, so 32 is outputted next. Finally, the class is loaded with the instance initialization blocks again being called before the constructor, outputting 45. Overloaded and overridden methods always have the same method name. The output is "5". At runtime, the object is passed around and, due to polymorphism, can be read using any of those references since the underlying object is a SoccerBall. In other words, casting it to a different reference variable does not modify the object or cause it to lose its underlying SoccerBall information. A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a method. The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. C The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because the class inherits two default methods with the same signature and no overridden version. The code does not compile because the class inherits two default methods with the same signature and no overridden	26	В	
happens first. The instance initialization blocks are executed before the constructor, so 32 is outputted next. Finally, the class is loaded with the instance initialization blocks again being called before the constructor, outputting 45. 28	27	C	
ouiputed next. Finally, the class is loaded with the instance initialization blocks again being called before the constructor, outputting 45. C Overloaded and overridden methods always have the same method name. The output is "5". At runtime, the object is passed around and, due to polymorphism, can be read using any of those references since the underlying object is a SoccerBall. In other words, casting it to a different reference variable does not modify the object or cause it to lose its underlying SoccerBall information. A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a method. The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. C The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A concret class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. The code does not compile because the class inherits two default methods with the same signature and no overridden version. The code does not compile because the class extends another class. The output is "2". The ref	1		outputted first. When the BlueCar is created in the main() method, the superclass initialization
before the constructor, outputting 45. C			happens first. The instance initialization blocks are executed before the constructor, so 32 is
29 A The output is "5". At runtime, the object is passed around and, due to polymorphism, can be read using any of those references since the underlying object is a SoccerBall. In other words, casting it to a different reference variable does not modify the object or cause it to lose its underlying SoccerBall information. 30 C A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a method. 31 B The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. 32 C The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. 33 B A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. 34 C A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. 35 D The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because the class inherits two default methods with the same signature and no overridden version. 36 B They both can contain default methods. 37 C The code does not compile because the class inherits two default methods with the same signature and no overridden version. 38 A Protected instance method is a virtual method. 40 A The output is "2". The reference type determines the version of the secret variable that is selected. 41 D Public final void dance() is a vaild override by a subclass. 42 C The answer is none of the above. 43 Final modifiers cannot be applied to an int			
The output is "5". At runtime, the object is passed around and, due to polymorphism, can be read using any of those references since the underlying object is a SoccerBall. In other words, casting to a different reference variable does not modify the object or cause it to lose its underlying SoccerBall information. C A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a method. The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. C The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class. B They both can contain default methods. C The code does not compile because it extends the Bird class. A Protected instance method is a virtual method. A The output is "2". The reference type determines the version of the secret variable that is selected. The answer is none of the above. The object type determines which attributes exist in memory, while the reference type determines which attributes exist in memory, while the reference type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. The object type de			
using any of those references since the underlying object is a SoccerBall. In other words, casting it to a different reference variable does not modify the object or cause it to lose its underlying SoccerBall information. C A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a method. The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. The code does not compile because the class inherits two default methods with the same signature and no overridden version. A Protected instance method is a virtual method. A Protected instance method is a virtual method. A The output is "2". The reference type determines the version of the secret variable that is selected. Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. The aparent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist			
to a different reference variable does not modify the object or cause it to lose its underlying SoccerBall information. A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a method. The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A class may be assigned to a subclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. They both can contain default methods. The code does not compile because the class inherits two default methods with the same signature and no overridden version. A Protected instance method is a virtual method. A Protected instance method is a virtual method. A The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. C The answer is none of the above. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. B Ald backward compatibility to existing interfaces is the best reason	29	Α	
SoccerBall information. C A class that defines an instance variable with the same name as a variable in the parent class is referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a method. The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. They both can contain default methods. The code does not compile because the class inherits two default methods with the same signature and no overridden version. A Protected instance method is a virtual method. A The coted instance method is a virtual method. A The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. C The answer is none of the above. The answer is none of the above. The answer is none of the above. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. The object type determines which attributes exist in memory, while the reference type determines which at			
C			, , , , , , , , , , , , , , , , , , ,
referred to as hiding a variable, while a class that defines a static method with the same signature as a static method in a parent class is referred to as hiding a method. The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A cass may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. The code does not compile because it extends the Bird class. The code does not compile because the class inherits two default methods with the same signature and no overridden version. A Protected instance method is a virtual method. A Protected instance method is a virtual method. A The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. C The answer is none of the above. Final modifiers cannot be applied to an interface method. If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the versi	20	C	
a static method in a parent class is referred to as hiding a method. The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid. C The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. C The code does not compile because the class inherits two default methods with the same signature and no overridden version. A Protected instance method is a virtual method. A The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is new of the above. B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator, which attributes are accessible by the caller. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this	30		
The code does not compile due to line x2 because the override of getEqualSides() in Square is invalid.			
invalid. The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract.	31	В	
The code does not compile because The Rotorcraft class includes an abstract method, but the class itself is not marked abstract. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. The code does not compile because it extends the Bird class. The sple does not compile because the class inherits two default methods with the same signature and no overridden version. Protected instance method is a virtual method. A Protected instance method is a virtual method. A The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. A Final modifiers cannot be applied to an interface method. It compiled and at runtime prints Let's start the party! The answer is none of the above. If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method.			
itself is not marked abstract. A class may be assigned to a superclass reference variable automatically but requires an explicit cast when assigned to a subclass reference variable. A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. D The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. They both can contain default methods. The code does not compile because the class inherits two default methods with the same signature and no overridden version. A Protected instance method is a virtual method. A Protected instance method is a virtual method. A The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is none of the above. If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method.	32	С	
cast when assigned to a subclass reference variable. A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. They both can contain default methods. The code does not compile because the class inherits two default methods with the same signature and no overridden version. A Protected instance method is a virtual method. A Protected instance method is a virtual method. A The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is none of the above. If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this			
A concrete class is the first non-abstract subclass that is required to implement all of the inherited abstract methods. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. The code does not compile because it extends the Bird class. The code does not compile because the class inherits two default methods with the same signature and no overridden version. Protected instance method is a virtual method. A Protected instance method is a virtual method. A Interface extends another interface, while a class extends another class. A The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. A Final modifiers cannot be applied to an interface method. A It compiled and at runtime prints Let's start the party! D The answer is none of the above. A If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this	33	В	A class may be assigned to a superclass reference variable automatically but requires an explicit
abstract methods. The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. The young to contain default methods. The code does not compile because the class inherits two default methods with the same signature and no overridden version. Protected instance method is a virtual method. An interface extends another interface, while a class extends another class. The output is "2". The reference type determines the version of the secret variable that is selected. Public final void dance() is a valid override by a subclass. The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is none of the above. It compiled and at runtime prints Let's start the party! The answer is none of the above. If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this			
The code contains three errors. The method fly() defined in CanFly is not marked static or default and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. They both can contain default methods. The code does not compile because the class inherits two default methods with the same signature and no overridden version. Protected instance method is a virtual method. An interface extends another interface, while a class extends another class. The output is "2". The reference type determines the version of the secret variable that is selected. Public final void dance() is a valid override by a subclass. The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is none of the above. If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. And backward compatibility to existing interfaces is the best reason for creating a default interface method. Chapter of the secret variable static production of the secret variable that is selected. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. And backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this	34	C	A concrete class is the first non-abstract subclass that is required to implement all of the inherited
and defines an implementation, an empty {}, meaning it cannot be assumed to be abstract. The implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. The both can contain default methods. The code does not compile because the class inherits two default methods with the same signature and no overridden version. A Protected instance method is a virtual method. A Interface extends another interface, while a class extends another class. A The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is none of the above. It compiled and at runtime prints Let's start the party! The answer is none of the above. If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this			
implementation of fly(int speed) in the Bird class also does not compile because of the signature. The Eagle class does not compile because it extends the Bird class. They both can contain default methods. The code does not compile because the class inherits two default methods with the same signature and no overridden version. A Protected instance method is a virtual method. A Protected instance method is a virtual method. A In interface extends another interface, while a class extends another class. The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. A Final modifiers cannot be applied to an interface method. It compiled and at runtime prints Let's start the party! The answer is none of the above. B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this	35	D	
The Eagle class does not compile because it extends the Bird class. They both can contain default methods. The code does not compile because the class inherits two default methods with the same signature and no overridden version. A Protected instance method is a virtual method. An interface extends another interface, while a class extends another class. The output is "2". The reference type determines the version of the secret variable that is selected. Public final void dance() is a valid override by a subclass. The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is new Wolf() because Wolf is not a sub class of Dog. The answer is none of the above. If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this			
36BThey both can contain default methods.37CThe code does not compile because the class inherits two default methods with the same signature and no overridden version.38AProtected instance method is a virtual method.39BAn interface extends another interface, while a class extends another class.40AThe output is "2". The reference type determines the version of the secret variable that is selected.41DPublic final void dance() is a valid override by a subclass.42CThe answer is new Wolf() because Wolf is not a sub class of Dog.43AFinal modifiers cannot be applied to an interface method.44AIt compiled and at runtime prints Let's start the party!45DThe answer is none of the above.46BIf a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition.47DThe object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller.48ALong is a subclass of Number, and therefore, it is covariant with the version in MusicCreator.49BAdd backward compatibility to existing interfaces is the best reason for creating a default interface method.50CThe code does not compile. Since IO Exception is a superclass of EOFException, we see that this			
The code does not compile because the class inherits two default methods with the same signature and no overridden version. A Protected instance method is a virtual method. A Interface extends another interface, while a class extends another class. A The output is "2". The reference type determines the version of the secret variable that is selected. Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. A Final modifiers cannot be applied to an interface method. It compiled and at runtime prints Let's start the party! The answer is none of the above. B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this	26	D	
and no overridden version. A Protected instance method is a virtual method. An interface extends another interface, while a class extends another class. A The output is "2". The reference type determines the version of the secret variable that is selected. Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. A Final modifiers cannot be applied to an interface method. It compiled and at runtime prints Let's start the party! The answer is none of the above. B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this			
38AProtected instance method is a virtual method.39BAn interface extends another interface, while a class extends another class.40AThe output is "2". The reference type determines the version of the secret variable that is selected.41DPublic final void dance() is a valid override by a subclass.42CThe answer is new Wolf() because Wolf is not a sub class of Dog.43AFinal modifiers cannot be applied to an interface method.44AIt compiled and at runtime prints Let's start the party!45DThe answer is none of the above.46BIf a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition.47DThe object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller.48ALong is a subclass of Number, and therefore, it is covariant with the version in MusicCreator.49BAdd backward compatibility to existing interfaces is the best reason for creating a default interface method.50CThe code does not compile. Since IO Exception is a superclass of EOFException, we see that this	3/		
 B An interface extends another interface, while a class extends another class. A The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. A Final modifiers cannot be applied to an interface method. A It compiled and at runtime prints Let's start the party! D The answer is none of the above. B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. D The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this 	39	Α	
 A The output is "2". The reference type determines the version of the secret variable that is selected. D Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. A Final modifiers cannot be applied to an interface method. A It compiled and at runtime prints Let's start the party! D The answer is none of the above. B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. D The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this 			
 D Public final void dance() is a valid override by a subclass. C The answer is new Wolf() because Wolf is not a sub class of Dog. A Final modifiers cannot be applied to an interface method. A It compiled and at runtime prints Let's start the party! D The answer is none of the above. B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. D The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. B Add backward compatibility to existing interfaces is the best reason for creating a default interface method. C The code does not compile. Since IO Exception is a superclass of EOFException, we see that this 		_	
 C The answer is new Wolf() because Wolf is not a sub class of Dog. A Final modifiers cannot be applied to an interface method. A It compiled and at runtime prints Let's start the party! D The answer is none of the above. B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. D The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. B Add backward compatibility to existing interfaces is the best reason for creating a default interface method. C The code does not compile. Since IO Exception is a superclass of EOFException, we see that this 			
A Final modifiers cannot be applied to an interface method. 44 A It compiled and at runtime prints Let's start the party! 45 D The answer is none of the above. 46 B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. 47 D The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. 48 A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. 49 B Add backward compatibility to existing interfaces is the best reason for creating a default interface method. 50 C The code does not compile. Since IO Exception is a superclass of EOFException, we see that this			
 A It compiled and at runtime prints Let's start the party! D The answer is none of the above. B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. D The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. B Add backward compatibility to existing interfaces is the best reason for creating a default interface method. C The code does not compile. Since IO Exception is a superclass of EOFException, we see that this 			
 D The answer is none of the above. B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. D The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. B Add backward compatibility to existing interfaces is the best reason for creating a default interface method. C The code does not compile. Since IO Exception is a superclass of EOFException, we see that this 			
 B If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition. D The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. B Add backward compatibility to existing interfaces is the best reason for creating a default interface method. C The code does not compile. Since IO Exception is a superclass of EOFException, we see that this 			
compiler), a child class must contain at least one constructor definition. The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. C The code does not compile. Since IO Exception is a superclass of EOFException, we see that this			
The object type determines which attributes exist in memory, while the reference type determines which attributes are accessible by the caller. Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. Add backward compatibility to existing interfaces is the best reason for creating a default interface method. The code does not compile. Since IO Exception is a superclass of EOFException, we see that this		1	
which attributes are accessible by the caller. 48 A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. 49 B Add backward compatibility to existing interfaces is the best reason for creating a default interface method. 50 C The code does not compile. Since IO Exception is a superclass of EOFException, we see that this	47	D	
 A Long is a subclass of Number, and therefore, it is covariant with the version in MusicCreator. B Add backward compatibility to existing interfaces is the best reason for creating a default interface method. C The code does not compile. Since IO Exception is a superclass of EOFException, we see that this 			
 B Add backward compatibility to existing interfaces is the best reason for creating a default interface method. C The code does not compile. Since IO Exception is a superclass of EOFException, we see that this 	48		
method. 50 C The code does not compile. Since IO Exception is a superclass of EOFException, we see that this			
			method.
is a broader exception and therefore not compatible.	50	C	
			is a broader exception and therefore not compatible.

Reference

Scott S, Jeanne B.: OCA/OCP Java SE 8 Programmer Practice Tests. Indiana, USA: 2017.