socrative

Answer Key
February 24, 2021

Java OCA Chapter 1 Review ('Copy')

1. Which of the following are valid Java identifiers? (Choose all that apply)

- ✓ **A** A$B
- ✓ **B** _helloWorld
- **C** true
- **D** java.lang
- ✓ **E** Public
- **F** 1980_s

i A, B, E. Option A is valid because you can use the dollar sign in identifiers. Option B is valid because you can use an underscore in identifiers. Option C is not a valid identifier because true is a Java reserved word. Option D is not valid because the dot (.) is not allowed in identifiers. Option E is valid because Java is case sensitive, so Public is not a reserved word and therefore a valid identifier. Option F is not valid because the first character is not a letter, $, or _.

2. What is the output of the following program? 1: public class WaterBottle { 2: private String brand; 3: private boolean empty; 4: public static void main(String[] args) { 5: WaterBottle wb = new WaterBottle(); 6: System.out.print("Empty = " + wb.empty); 7: System.out.print(", Brand = " + wb.brand); 8: } }

- **A** Line 6 generates a compiler error.
- **B** Line 7 generates a compiler error.
- **C** There is no output.
- ✓ **D** Empty = false, Brand = null
- **E** Empty = false, Brand =
- **F** Empty = null, Brand = null

i Boolean fields initialize to false and references initialize to null, so Empty is false and Brand is null. Brand = null is output.

3. Which of the following are true? (Choose all that apply) 4: short numPets = 5; 5: int numGrains = 5.6; 6: String name = "Scruffy"; 7: numPets.length(); 8: numGrains.length(); 9: name.length();

- A Line 4 generates a compiler error
- ✓ B Line 5 generates a compiler error
- C Line 6 generates a compiler error
- ✓ D Line 7 generates a compiler error
- ✓ E Line 8 generates a compiler error
- F Line 9 generates a compiler error
- G The code compiles as is

i Option A (line 4) compiles because short is an integral type. Option B (line 5) generates a compiler error because int is an integral type, but 5.6 is a floating-point type. Option C (line 6) compiles because it is assigned a String. Options D and E (lines 7 and 8) do not compile because short and int are primitives. Primitives do not allow methods to be called on them. Option F (line 9) compiles because length() is defined on String.

4. Given the following class, which of the following is true? (Choose all that apply) 1: public class Snake { 2: 3: public void shed(boolean time) { 4: 5: if (time) { 6: 7: } 8: System.out.println(result); 9: 10: } 11: }

- ✓ A If String result = "done"; is inserted on line 2, the code will compile.
- ✓ B If String result = "done"; is inserted on line 4, the code will compile.
- C If String result = "done"; is inserted on line 6, the code will compile.
- D If String result = "done"; is inserted on line 9, the code will compile.
- E None of the above changes will make the code compile.

i Adding the variable at line 2 makes result an instance variable. Since instance variables are in scope for the entire life of the object, option A is correct. Option B is correct because adding the variable at line 4 makes result a local variable with a scope of the whole method. Adding the variable at line 6 makes result a local variable with a scope of lines 6–7. Since it is out of scope on line 8, the println does not compile and option C is incorrect. Adding the variable at line 9 makes result a local variable with a scope of lines 9 and 10. Since line 8 is before the declaration, it does not compile and option D is incorrect. Finally, option E is incorrect because the code can be made to compile.

5. Given the following classes, which of the following can independently replace INSERT IMPORTS HERE to make the code compile? (Choose all that apply.) package aquarium; public class Tank { } package aquarium.jellies; public class Jelly { } package visitor; INSERT IMPORTS HERE public class AquariumVisitor { public void admire(Jelly jelly) { } }

- A import aquarium.*;
- B import aquarium.*.Jelly;
- ✓ C import aquarium.jellies.Jelly;
- ✓ D import aquarium.jellies.*;
- E import aquarium.jellies.Jelly.*;
- F None of these can make the code compile.

i Option C is correct because it imports Jelly by classname. Option D is correct because it imports all the classes in the jellies package, which includes Jelly. Option A is incorrect because it only imports classes in the aquarium package—Tank in this case—and not those in lower-level packages. Option B is incorrect because you cannot use wildcards anyplace other than the end of an import statement. Option E is incorrect because you cannot import parts of a class with a regular import statement. Option F is incorrect because options C and D do make the code compile.

6. Given the following classes, what is the maximum number of imports that can be removed and have the code still compile? package aquarium; public class Water { } package aquarium; import java.lang.*; import java.lang.System; import aquarium.Water; import aquarium.*; public class Tank { public void print(Water water) { System.out.println(water); } }

- A 0
- B 1
- C 2
- D 3
- ✓ E 4
- F Does not compile

i The first two imports can be removed because java.lang is automatically imported. The second two imports can be removed because Tank and Water are in the same package, making the correct answer E. If Tank and Water were in different packages, one of these two imports could be removed. In that case, the answer would be option D.

7. Given the following classes, which of the following snippets can be inserted in place of INSERT IMPORTS HERE and have the code compile? (Choose all that apply) package aquarium; public class Water { boolean salty = false; } package aquarium.jellies; public class Water { boolean salty = true; } package employee; INSERT IMPORTS HERE public class WaterFiller { Water water; }

- ✓ **A** import aquarium.*;
- ✓ **B** import aquarium.Water; import aquarium.jellies.*;
- ✓ **C** import aquarium.*; import aquarium.jellies.Water;
- **D** import aquarium.*; import aquarium.jellies.*;
- **E** import aquarium.Water; import aquarium.jellies.Water;
- **F** None of these imports can make the code compile.

ℹ Option A is correct because it imports all the classes in the aquarium package including aquarium.Water. Options B and C are correct because they import Water by classname. Since importing by classname takes precedence over wildcards, these compile. Option D is incorrect because Java doesn't know which of the two wildcard Water classes to use. Option E is incorrect because you cannot specify the same classname in two imports.

8. Given the following class, which of the following calls print out Blue Jay? public class BirdDisplay { public static void main(String[] name) { System.out.println(name[1]); } }

- **A** java BirdDisplay Sparrow Blue Jay
- ✓ **B** java BirdDisplay Sparrow "Blue Jay"
- **C** java BirdDisplay Blue Jay Sparrow
- **D** java BirdDisplay "Blue Jay" Sparrow
- **E** java BirdDisplay.class Sparrow "Blue Jay"
- **F** java BirdDisplay.class "Blue Jay" Sparrow
- **G** Does not compile

ℹ Option B is correct because arrays start counting from zero and strings with spaces must be in quotes. Option A is incorrect because it outputs Blue. C is incorrect because it outputs Jay. Option D is incorrect because it outputs Sparrow. Options E and F are incorrect because they output Error: Could not find or load main class BirdDisplay.class.

9. Which of the following legally fill in the blank so you can run the main() method from the command line? (Choose all that apply.) public static void main(_____)

- ✓ **A** String[] _names
- **B** String[] 123
- ✓ **C** String abc[]
- ✓ **D** String _Names[]
- ✓ **E** String... $n
- **F** String names
- **G** None of the above

i Option A is correct because it is the traditional main() method signature and variables may begin with underscores. Options C and D are correct because the array operator may appear after the variable name. Option E is correct because varargs are allowed in place of an array. Option B is incorrect because variables are not allowed to begin with a digit. Option F is incorrect because the argument must be an array or varargs. Option F is a perfectly good method. However, it is not one that can be run from the command line because it has the wrong parameter type.

10. Which of the following are legal entry point methods that can be run from the command line? (Choose all that apply)

- **A** private static void main(String[] args)
- **B** public static final main(String[] args)
- **C** public void main(String[] args)
- **D** public static void test(String[] args)
- ✓ **E** public static void main(String[] args)
- **F** public static main(String[] args)
- **G** None of the above

i Option E is the canonical main() method signature. You need to memorize it. Option A is incorrect because the main() method must be public. Options B and F are incorrect because the main() method must have a void return type. Option C is incorrect because the main() method must be static. Option D is incorrect because the main() method must be named main.

11. Which of the following are true? (Choose all that apply)

- **A** An instance variable of type double defaults to null.
- **B** An instance variable of type int defaults to null.
- ✓ **C** An instance variable of type String defaults to null.
- ✓ **D** An instance variable of type double defaults to 0.0.
- **E** An instance variable of type int defaults to 0.0.
- **F** An instance variable of type String defaults to 0.0.
- **G** None of the above

Option C is correct because all non-primitive values default to null. Option D is correct because float and double primitives default to 0.0. Options B and E are incorrect because int primitives default to 0.

12. Which of the following are true? (Choose all that apply)

- A A local variable of type boolean defaults to null.
- B A local variable of type float defaults to 0.
- C A local variable of type Object defaults to null.
- D A local variable of type boolean defaults to false.
- E A local variable of type boolean defaults to true.
- F A local variable of type float defaults to 0.0.
- ✓ G None of the above

Option G is correct because local variables do not get assigned default values. The code fails to compile if a local variable is not explicitly initialized. If this question were about instance variables, options D and F would be correct. A boolean primitive defaults to false and a float primitive defaults to 0.0.

13. Which of the following are true? (Choose all that apply)

- ✓ A An instance variable of type boolean defaults to false.
- B An instance variable of type boolean defaults to true.
- C An instance variable of type boolean defaults to null.
- ✓ D An instance variable of type int defaults to 0.
- E An instance variable of type int defaults to 0.0.
- F An instance variable of type int defaults to null.
- G None of the above

Option A is correct because boolean primitives default to false and int primitives default to 0.

14. Given the following class in the file /my/directory/named/A/Bird.java: INSERT CODE HERE public class Bird { } Which of the following replaces INSERT CODE HERE if we compile from /my/directory? (Choose all that apply)

- A package my.directory.named.a;
- B package my.directory.named.A;
- C package named.a;
- ✓ D package named.A;
- E package a;
- F package A;
- G Does not compile

The package name represents any folders underneath the current path, which is named.A in this case. Option B is incorrect because package names are case sensitive, just like variable names and other

identifiers.

15. Which of the following lines of code compile? (Choose all that apply)

- ✓ **A** int i1 = 1_234;
- **B** double d1 = 1_234_.0;
- **C** double d2 = 1_234._0;
- **D** double d3 = 1_234.0_;
- ✓ **E** double d4 = 1_234.0;
- **F** None of the above

i Underscores are allowed as long as they are directly between two other digits. This means options A and E are correct. Options B and C are incorrect because the underscore is adjacent to the decimal point. Option D is incorrect because the underscore is the last character.

16. Given the following class, which of the following lines of code can replace INSERT CODE HERE to make the code compile? (Choose all that apply). public class Price { public void admission() { // INSERT CODE HERE System.out.println(amount); } }

- **A** int amount = 9L;
- ✓ **B** int amount = 0b101;
- ✓ **C** int amount = 0xE;
- ✓ **D** double amount = 0xE;
- **E** double amount = 1_2_.0_0;
- **F** int amount = 1_2_;
- **G** None of the above

i 0b is the prefix for a binary value and is correct. 0X is the prefix for a hexadecimal value. This value can be assigned to many primitive types, including int and double, making options C and D correct. Option A is incorrect because 9L is a long value. long amount = 9L would be allowed. Option E is incorrect because the underscore is immediately before the decimal. Option F is incorrect because the underscore is the very last character.

17. Which of the following are true? (Choose all that apply) public class Bunny { public static void main(String[] args) { Bunny bun = new Bunny(); } }

- ✓ **A** Bunny is a class.
- **B** bun is a class.
- **C** main is a class.
- **D** Bunny is a reference to an object.
- ✓ **E** bun is a reference to an object.
- **F** main is a reference to an object.
- **G** None of the above

i Bunny is a class, which can be seen from the declaration: public class Bunny. bun is a reference to an object. main() is a method.

18. Which represent the order in which the following statements can be assembled into a program that will compile successfully? (Choose all that apply) A: class Rabbit {} B: import java.util.*; C: package animals;

A A, B, C

B B, C, A

✓ C C, B, A

✓ D B, A

✓ E C, A

F A, C

G A, B

i package and import are both optional. If both are present, the order must be package, then import, then class. Option A is incorrect because class is before package and import. Option B is incorrect because import is before package. Option F is incorrect because class is before package. Option G is incorrect because class is before import.

19. Suppose we have a class named Rabbit. Which of the following statements are true? (Choose all that apply) 1: public class Rabbit { 2: public static void main(String[] args) { 3: Rabbit one = new Rabbit(); 4: Rabbit two = new Rabbit(); 5: Rabbit three = one; 6: one = null; 7: Rabbit four = one; 8: three = null; 9: two = null; 10: two = new Rabbit(); 11: System.gc(); 12: } }

A The Rabbit object from line 3 is first eligible for garbage collection immediately following line 6.

✓ B The Rabbit object from line 3 is first eligible for garbage collection immediately following line 8.

C The Rabbit object from line 3 is first eligible for garbage collection immediately following line 12.

✓ D The Rabbit object from line 4 is first eligible for garbage collection immediately following line 9.

E The Rabbit object from line 4 is first eligible for garbage collection immediately following line 11.

F The Rabbit object from line 4 is first eligible for garbage collection immediately following line 12.

i The Rabbit object from line 3 has two references to it: one and three. The references are nulled out on lines 6 and 8, respectively. Option B is correct because this makes the object eligible for garbage collection after line 8. Line 7 sets the reference four to the now null one, which means it has no effect on garbage collection. The Rabbit object from line 4 only has a single reference to it: two. Option D is correct because this single reference becomes null on line 9. The Rabbit object declared on line 10 becomes eligible for garbage collection at the end of the method on line 12. Calling System.gc() has no effect on eligibility for garbage collection.

20. What is true about the following code? (Choose all that apply) public class Bear { protected void finalize() { System.out.println("Roar!"); } public static void main(String[] args) { Bear bear = new Bear(); bear = null; System.gc(); } }

- A finalize() is guaranteed to be called.
- ✓ B finalize() might or might not be called
- C finalize() is guaranteed not to be called.
- D Garbage collection is guaranteed to run.
- ✓ E Garbage collection might or might not run.
- F Garbage collection is guaranteed not to run.
- G The code does not compile.

i Calling System.gc() suggests that Java might wish to run the garbage collector. Java is free to ignore the request, making option E correct. finalize() runs if an object attempts to be garbage collected, making option B correct.

21. What does the following code output? 1: public class Salmon { 2: int count; 3: public void Salmon() { 4: count = 4; 5: } 6: public static void main(String[] args) { 7: Salmon s = new Salmon(); 8: System.out.println(s.count); 9: } }

- ✓ A 0
- B 4
- C Compilation fails on line 3.
- D Compilation fails on line 4.
- E Compilation fails on line 7.
- F Compilation fails on line 8.

i While the code on line 3 does compile, it is not a constructor because it has a return type. It is a method that happens to have the same name as the class. When the code run, the default constructor is called and count has the default value (0) for an int.

22. Which of the following are true statements? (Choose all that apply)

- A Java allows operator overloading.
- ✓ B Java code compiled on Windows can run on Linux.
- C Java has pointers to specific locations in memory.
- D Java is a procedural language.
- ✓ E Java is an object-oriented language.
- F Java is a functional programming language.

i C++ has operator overloading and pointers. Java made a point of not having either. Java does have references to objects, but these are pointing to an object that can move around in memory. Option B is correct because Java is platform independent. Option E is correct because Java is object oriented. While it does support some parts of functional programming, these occur within a class.