OCA Java Test

1. **What is result of the following code snippet?(Choose all that apply)**

3: **final** **int** movieRating=4;

4: **int** badMovie=0;

5: **switch**(badMovie){

6: **case** 0:

7: **case** badMovie:System.***out***.println("Awful");**break**;

8: **case** 4:

9: **case** movieRating:Systtem.out.println("Great"); **break**;

10: **default**:

11: **case**(**int**)'a':

12: **case** 1\*1: System.***out***.println("Too be determined");**break**;

13: }

**○** A)The code will not compile because of line 6.

**○** B)The code will not compile because of line 7.

**○** C)The code will not compile because of line 8.

**○** D)The code will not compile because of line 9.

**○** E)The code will not compile because of line 10.

**○** F)The code will not compile because of line 11.

1. **What is the result of the following ?**

List<String>fish=**new** ArrayList<>();

fish.add("goldfish");

fish.add("minnow");

fish.remove("goldfish");

fish.remove("shark");

System.***out***.println(fish.size());

**○** A) 0

**○** B) 1

**○** C) 2

**○** D) An exception is throw.

**○** E) The code does not compile.

1. **What is the result of the following ?**

3: **int**[] times [] =**new** **int**[3][3];

4: **for**(**int** i=0 ; i < times.length ; i++)

5: **for**(**int** j=0 ;j < times.length ; j++)

6: times[i,j]=i\*j;

7: System.out.println(times[2,2]);

**○** A) 1

**○** B) 4

**○** C) 1 printed 4 times

**○** D) 4 printed 3 times

**○** E) An exception is thrown .

**○** F) The code fails to compile because of line 3.

**○** G) The code fails to compile for another reason.

1. **What is the output when new Buffalo() is called?**

1: **public** **class** Animal {

2: **public** Animal(**int** arg) { System.***out***.println("1"); **this**(); }

3: **public** Animal() { System.***out***.println("2"); }

4: }

5: **public** **class** Buffalo **extends** Animal {

6: **public** Buffalo() {

7: System.***out***.println("3");

8: }

9: }

**○** A) 123

**○** B) 213

**○** C) 3

**○** D) 23

**○** E) The code will not compile because of line 2.

**○** F) The code will not compile because of line 7.

**○** G) It compiles but throws an exception at runtime.

1. **What is the output of the following code?**

1: **interface** WaterFowl {}

2: **class** Bird {}

3: **public** **class** Duck **extends** Bird **implements** WaterFowl {

4: **public** **void** quack() { System.***out***.println("quack!"); }

5: **public** **static** **void** main(String[] args) {

6: Object object=(Object)**new** Bird();

7: Duck duck = (Duck)object;

8: duck.quack();

9: }

10: }

**○** A) quack!

**○** B) No output

**○** C) The code will not compile because of line 4.

**○** D) The code will not compile because of line 6.

**○** E) The code will not compile because of line 7.

**○** F) The code compiles but throws an exception at runtime.

1. **Which lambda can replace the MySecret class?(Choose all that apply)**

**interface** Secret{

String test(String a, String b);

}

**class** MySecret **implements** Secret {

**public** String test(String a, String b) {

**return** a + b;

}

}

**○** A) caller((a, b) -> a + b);

**○** B) caller((String a, b) -> a + b);

**○** C) caller((String a, String b) -> a + b);

**○** D) caller((a, b) , a + b);

**○** E) caller((String a, b) , a + b);

**○** F) caller((String a, String b) , a + b);

1. **Which lambda can replace the MySecret class?(Choose all that apply)**

**interface** Secret{

**int** number();

}

**class** MySecret **implements** Secret {

**public** **int** number() {

**return** 5;

}

}

**○** A) caller (-> 5);

**○** B) caller (x -> 5);

**○** C) caller (() -> 5);

**○** D) caller ((x) -> 5);

**○** E) caller (() -> {5});

**○** F) caller (() -> {return 5;});

1. **What is the result of the following program?**

1: **public** **class** Egret {

2: **private** String color;

3: **public** Egret (String color){

4: color = color;

5: }

6: **public** **static** **void** main(String[] args) {

7: Egret e = **new** Egret();

8: System.***out***.println("Color:" + e.color);

9: } }

**○** A) Color:

**○** B) Color: null

**○** C) Color: White

**○** D) Compiler error on line 4

**○** E) Compiler error on line 7

**○** F) Compiler error another line

1. **What is the result of this code?**

**public** **class** Chicken {

**private** **void** layEggs(**int**... eggs) {

System.***out***.print("many " + eggs[0] + " ");

}

**private** **void** layEggs(**int** eggs) {

System.***out***.print("one " + eggs + " ");

}

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

Chicken c=**new** Chicken();

c.layEggs(1, 2);

c.layEggs(3);

c.layEggs(**new** Integer(2));

}

}

**○** A) many 1 one 3 one 2

**○** B) many 2 one 3 one 2

**○** C) many 1 many 3 one 2

**○** D) many 2 one 3 one 2

**○** E) The code does not compile.

**○** F) An exception is thrown .

1. **What is the output of the following code?**

1: **public** **class** Fish {

2: **private** **static** String getColor() { **return** "Yellow"; }

3: **public** **static** **void** main(String[] args) {

4: **new** Pufferfish().printDescription();

5: }

6: }

7: **class** Pufferfish **extends** Fish {

8: **protected** **static** String getColor() { **return** "Green"; }

9: **public** **void** printDescription() {

10: System.***out***.println(**super**.*getColor*()+","+**this**.*getColor*()+","+*getColor*());

11: }

12: }

**○** A) Yellow, Green, Green

**○** B) Yellow, Green, Yellow

**○** C) Green, Yellow, Green

**○** D) Green, Green, Green

**○** E) The code will not compile because of line 8.

**○** F) The code will not compile because of line 10.

**○** G) The code compiles but throws an exception at runtime.

1. **What does the following code output when run as java Duck Duck Goose?**

**package** oca;**public** **class** Duck {

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

**for**(**int** i=1; i<+ args.length; i++)

System.***out***.println(args[i]);

}

}

**○** A) Duck Goose

**○** B) Duck ArrayIndexOutOfBoundsException

**○** C) Goose

**○** D) Goose ArrayIndexOutOfBoundsException

**○** E) None of the above

1. **Which of the following complete this method to print out all the arguments?(Choose all that apply)**

**public** **void** print(String... args) {

//INSERT CODE HERE

}

**○** A) for (int i=0; i< args.length; i++)

System.out.println(args[i]);

**○** B) for (int i=0; i<= args.length; i++)

System.out.println(args[i]);

**○** C) for (int i=1; i< args.length; i++)

System.out.println(args[i]);

**○** D) for (int i=1; i<= args.length; i++)

System.out.println(args[i]);

**○** E) for (String arg : args)

System.out.println(arg);

**○** F) None of the above; args is not an array so it cannot be accessed like one.

1. **What is the output of the following snippet?**

**int** a=123;

**int** b=0;

**try**{

System.***out***.print(a / b);

System.***out***.print("1");

}**catch**(ArithmeticException e) {

System.***out***.print("2");

}**catch** (RuntimeException r) {

System.***out***.print("3");

}**finally** {

System.***out***.print("4");

}

**○** A)14

**○** B) 2

**○** C) 24

**○** D) 234

**○** E) The code does not compile.

**○** F) An uncaught exception is thrown.

1. **Which of the following statements about objects and references are true in Java? (Choose all that apply)**

**○** A) By explicitly casting an object to a subclass, you can gain access to methods and variables that were hidden from access.

**○** B) If you compare two distinct references to the same object with==, the result will evaluate to false.

**○** C) All objects in memory can be referenced using a reference of type java.lang.Object.

**○** D) When you create an object in Java, you get Direct access to the object in memory.

**○** E) Removing all references to an object deletes the object from memory.

1. **Which of the following can fill in the blank to make this code compile?(Choose all that apply)**

**public** **class** ExceptionLogger {

**public** **static** **void** logException(RuntimeException e) {

System.***out***.println("Logging "+ e);

}

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

*logException*(**new** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_);

}

}

**○** A) Error()

**○** B) Exception()

**○** C) java.io.IOException()

**○** D) NullPointerException()

**○** E) RuntimeException()

1. **What is the result of the following code?**

1: **interface** Climb {

2: **boolean** isTooHigh( **int** height, **int** limit);

3: }

4:

5: **public** **class** Climber {

6: **public** **static** **void** main(String[] args) {

7: *check*((h,l) -> h > l,5);

8: }

9: **private** **static** **void** check(Climb climb, **int** height ){

10: **if** (climb.isTooHigh(height, "max"))

11: System.***out***.println("too high");

12: **else**

13: System.***out***.println("ok");

14: }

15: }

**○** A) ok

**○** B) too high

**○** C) Compiler error on line 7

**○** D) Compiler error on line 10

**○** E) Compiler error on a different line

**○** F) A runtime exception is thrown.

1. **What is the result of the following code?**

StringBuilder b = **new** StringBuilder();

b.append("maybe").insert(2, "sh").insert(5, "xy");

System.***out***.println(b.toString());

**○** A) mashyxybe

**○** B) mashybexy

**○** C) mayshbxye

**○** D) mayshxybe

**○** E) An exception is thrown.

**○** F) The code does not compile.

1. **Which statements are true for a traditional try statement (not a try-with-resources statement)?(Choose all that apply)**

**○** A) If a try statement has a catch clause, it is required to have finally clause.

**○** B) If a try statement does not have a catch clause, it is required to have a finally clause.

**○** C) If a try statement has a finally clause, it is required to have exactly one catch clause.

**○** D) If a try statement has a finally clause, it is required to have at least one catch clause.

**○** E) A try statement can exist without a catch or finally clause.

**○** F) A try statement can have both a catch and finally clause.

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

**○** A) do while loops can execute the loop body exactly zero times.

**○** B) do while loops contain an increment clause.

**○** C) for loops can execute the loop body exactly zero times.

**○** D) for loops contain an increment clause.

**○** E) while loops can execute the loop body exactly zero times.

**○** F) while loops contain an increment clause.

1. **What is the result of following code?**

**public** **class** Deer {

**static** **int** *count*;

**static** { *count*=0; }

Deer(){

*count*++;

}

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

*count*++;

Deer mother = **new** Deer();

Deer father= **new** Deer();

Deer fawn= **new** Deer();

System.***out***.println(father.*count*);

}

}

**○** A) 0

**○** B) 1

**○** C) 2

**○** D) 3

**○** E) 4

**○** F) 5

1. **What is the result of the following code?**

String numbers = "2468";

**int** total = 0;

total += numbers.indexOf("6");

total += numbers.length();

**char** ch = numbers.charAt(4);

System.***out***.println(total + " " + ch);

**○** A) 5 8

**○** B) 6 8

**○** C) 7 8

**○** D) 8 8

**○** E) An exception is thrown.

**○** F) The code does not compile.

1. **What is the result of following code?**

**public** **class** A {

**private** **int** i = 6;

**private** **int** j = i;

**public** A() {

i = 5;

}

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

A a = **new** A();

System.***out***.println(a.i + a.j);

}

}

**○** A) 55

**○** B) 56

**○** C) 66

**○** D) 10

**○** E) 11

**○** F) 12

**○** G) The code does not compile.

1. **What is the output when New BlackRhino() is called?**

**public class** Rhinoceros {

**public** Rhinoceros(){ System.***out***.print("1"); }

}

**public class** BlackRhino **extends** Rhinoceros {

**public** BlackRhino (**int** age) { System.***out***.print("2"); }

**public** BlackRhino() {

**this**(5);

System.***out***.print("3");

}

}

**○** A) 123

**○** B) 132

**○** C) 12

**○** D) 13

**○** E) This code does not compile.

1. **Given the following two classes, which of the following statements will compile when inserted on line 2?(Choose all that apply)**

1: **package** lilypad;

2: **public** **class** Frog { }

1: **package** lilypad;

2: // INSERT CODE HERE

3: **public** **class** Tadpole {

4: **private** Frog parent;

5: }

**○** A) import Frog;

**○** B) import lilypad.Frog;

**○** C) import lilipad.Tadpole;

**○** D) import static lilypad.Tadpole;

**○** E) static import lilypad.Tadepole;

**○** F) The code compiles without any code additions.

1. **What could be the output of the following code given that e() could be left with a blank implementation or have a one-line implementation that throws any type of exception?**

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

System.***out***.print("a");

**try**{

System.***out***.print("b");

*e*();

} **finally** {

System.***out***.print("c");

}

System.***out***.print("d");

}

**private** **static** **void** e() {

// code omitted

}

}

**○** A) ab

**○** B) abc

**○** C) abd

**○** D) abcd

**○** E) abc folowing by a stack trace

**○** F) None of the above; the code does not compile as is.

1. **Which of the following imports can be inserted to make the Robot class compile? (Choose all that apply)**

**package** util;

**public** **interface** Ports {

**int** ***RIGHT\_MOTOR*** = 1;

**int** ***LEFT\_MOTOR***= 2;

}

**package** robot;

// INSERT CODE HERE

**public** **class** Robot {

Robot ( **int** motor1, **int** motor2) { }

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

Robot robbie = **new** Robot (Ports.RIGHT\_MOTOR, LEFT\_MOTOR);

}

}

**○** A) import static util.Ports.\*;

**○** B) import static util.Ports.\*; import util.\*;

**○** C) import static util.Ports;

**○** D) static import util.Ports.\*;

**○** E) static import util.Ports.\*; import util.\*;

**○** F) static import util.Ports;

1. **Given the following code, which of the statements are true?(Choose all that apply)**

1: **package** animal;

2: **public** **class** Frog {

3: \_\_\_\_\_\_\_ **void** ribbit() { }

4: }

1: **package** \_\_\_\_\_\_;

2: **import** animal.\*;

3: **public** **class** Tadpole **extends** Frog {

4: **public** **static** **void** main(String[] args) {

5: Tadpole t = **new** Tadpole();

6: t.ribbit();

7: }

8: }

**○** A) If Tadpole is in package animal, t.ribbit() will compile if given default access.

**○** B) If Tadpole is in package animal, t.ribbit() will compile if given protected access.

**○** C) If Tadpole is in package animal, t.ribbit() will compile if given private access.

**○** D) If Tadpole is in package body, t.ribbit() will compile if given default access.

**○** E) If Tadpole is in package body, t.ribbit() will compile if given protected access.

**○** F) If Tadpole is in package body, t.ribbit() will compile if given private access.

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

**○** A) All String literals are automatically instantiated into a String object.

**○** B) The StringBuilder and StringBuffer classes define the same public methods.

**○** C) A StringBuilder object is immutable.

**○** D) A StringBuilder object cannot change its length once it is instantiated.

**○** E) A String object cannot change its length once it is instantiated.

1. **What is the result of this code?**

**public** **class** C {

String seq = "c";

{ seq += "g"; }

**public** C(){

**this** ("abc");

seq += "y";

}

**public** C(String s){

seq +="e";

}

{ seq += "z"; }

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

C c=**new** C();

System.***out***.println(c.seq);

}

}

**○** A) ceygz

**○** B) cgeyz

**○** C) cgyez

**○** D) cgzey

**○** E) cgzye

**○** F) cyegz

**○** G) The code does not compile.

1. **Which of the following lines do not compile?(Choose all that apply)**

1: **interface** Swim {

2: **default** **void** swim() { }

3: }

4: **class** Swimmer **implements** Swim {

5: **public** **void** crawl() {}

6: **default** **void** butterfly(**boolean** fast) {}

7: **private** **int** numberStrokes() **return** 0;

8: secret **void** secretWeapon() {}

9: }

**○** A) Line 2

**○** B) Line 5

**○** C) Line 6

**○** D) Line 7

**○** E) Line 8

**○** F) None; all the lines compile.

1. **In which order can you assemble the following statements in to make a valid Java program?(Choose all that apply)**

A: /\* hello world \*/

B: ArrayList list;

C: **import** java.util.\*;

D: **package** abc;

E: **public** **class** Builder {

F: **public** void method() {}

**○** A) A,D,E,B,F

**○** B) A,D,C,E,F,B

**○** C) D,A,E,B,F

**○** D) D,A,E,F,B

**○** E) D,C,A,E,B,F

**○** F) D,C,A,E,F,B

1. **Which of the following can fill in the blank to make the code compile?(Choose all that apply)**

**class** LimpException **extends** Exception{}

**class** HurtException **extends** LimpException {}

**public** **void** run(){

**try** {

split();

}**catch**(HurtException e) {

}**catch**(\_\_\_\_\_\_\_\_\_ e){

}

}

**private** **void** split() **throws** HurtException{ }

**○** A) Exception

**○**  B) HurtException

**○** C) IOException

**○** D) LimpException

**○** E) RuntimeException

1. **Which creates an object with the current hours, minutes, and seconds?**

**○** A) new LocalDate();

**○** B) new LocalTime();

**○** C) LocalDate.current();

**○** D) LocalTime.current();

**○** E) LocalDate.now();

**○** F) LocalTime.now();

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

**○** A) A StringBuilder can be cast to a String.

**○** B) A String can be passed to the constructor of a StringBuilder.

**○** C) StringBuilder is immutable.

**○** D) Stringbuilder is thread-safe.

**○** E) String is more efficient than StringBuilder when concatenating thousands of values.

**○** F) StringBuilder is more efficient than String when concatenating thousands of values.

1. **Which is true of the following code?(Choose all that apply)**

1: **package** abc;

2: **import** java.math.\*;

3: **public** **class** W {

4: **public** **void** method() {}

5: **int** a;

6:}

**○** A) The code will not compile if line 1 is removed.

**○** B) The code will not compile if line 2 is removed.

**○** C) The code will not compile if line 3 is removed.

**○** D) The code will not compile if line 4 is removed.

**○** E) The code will not compile if line 5 is removed.

1. **Which of the following are ordered from smallest to largest? (Choose all that apply)**

**○** A) byte, int, short, long

**○** B) byte, short, int, long

**○** C) short, byte, int, long

**○** D) short, int, byte, long

**○** E) double, float

**○** F) float, double

1. **What is the result of the following code?**

1: **public** **class** PrintIntegers {

2: **public** **static** **void** main(String[] args) {

3: **int** y = -2;

4: **do** System.***out***.print(++y + " ");

5: **while** (y <= 5);

6: } }

**○** A) -2 -1 0 1 2 3 4 5

**○** B) -2 -1 0 1 2 3 4

**○** C) -1 0 1 2 3 4 5 6

**○** D) -1 0 1 2 3 4 5

**○** E) The code will not compile because of line 5.

**○** F) The code contains an infinite loop and does not terminate.

1. **Which of the following lists is a correct sequence for catching exceptions? (Choose all that apply)**

**○** A) ArrayIndexOutOfBoundsException, IOException, RuntimeException

**○** B) ArrayIndexOutOfBoundsException, Exception, IOException

**○** C) Exception, IOException, ArrayIndexOutOfBoundsException

**○** D) Exception, ArrayIndexOutOfBoundsException, IOException

**○** E) IOException, RuntimeException, ArrayIndexOutOfBoundsException

**○** F) IOException, ArrayIndexOutOfBoundsException, Exception

1. **Which of the following lambda expressions can be passed to a function of Predicate<String>type?(Choose all that apply)**

**○** A) check(() -> s.isEmpty());

**○** B) check(s -> s.isEmpty());

**○** C) check(String s -> s.isEmpty());

**○** D) check((String s) -> s.isEmpty());

**○** E) check((s1) -> s.isEmpty());

**○** F) check((s1, s2) -> s.isEmpty());

1. **What is the result of following code?**

String s= "a";

s += 1;

s.concat(s);

s.concat(".");

System.***out***.println(s);

**○** A) a

**○** B) a1

**○** C) a1a

**○** D) a1a.

**○** E) An exception is thrown.

**○** F) The code does not compile.

1. **What is the result of the following?**

23: List<String> list = Arrays.*asList*("a", "B", "d", "c");

24: Collections.*sort*(list);

25: String[] array = list.toArray(**new** String[4]);

26: **for** (String string : array) System.***out***.print(string);

**○** A) aBcd

**○** B) aBdc

**○** C) Bacd

**○** D) Compiler error on line 9

**○** E) Compiler error on line 11

**○** F) An exception is thrown.

1. **Which of the following are true of the following code?(Choose all that apply)**

1: **public** **class** Giraffe {

2: **public** **int** height;

3: **public** **int** getHeight() {

4: **return** height;

5: }

6: **public** **void** setHeight(**int** height){

7: **this**.height = height;

8: }

9: }

**○** A) This class currently exhibits data encapsulation.

**○** B) This class currently exhibits immutability.

**○** C) Making line 2 private would exhibit data encapsulation.

**○** D) Making line 2 private would exhibit data immutability.

**○** E) Making line 2 private and removing lines 6-8 would exhibit data encapsulation.

**○** F) Making line 2 private and removing lines 6-8 would exhibit immutability.

1. **What is the result of this code?**

1: **public** **class** Dino {

2: **final** String species = "Triceratops";

3: **double** weight;

4: **public** Dino(**double** weight){

5: **this**.weight = weight;

6: species = "Raptor";

7: }

8: **public** **static** **void** main(String[] args) {

9: Dino dino = **new** Dino(500);

10: System.***out***.println(dino.weight);

11: } }

**○** A) Compiler error on line 5

**○** B) Compiler error on line 6

**○** C) Compiler error on line 9

**○** D) Compiler error on line 10

**○** E) 500

1. **Which exception will the following throw?**

**int**[] nums =**new** **int**[] { 1, 4, 6};

Object p= nums;

**int**[] two = (**int**[]) p;

System.***out***.println(two[two.length]);

**○** A) ArrayIndexOutOfBoundsException

**○** B) ClassCastException

**○** C) IllegalArgumentException

**○** D) NumberFormatException

**○** E) None of the above

1. **What could be the output of the following code given that e() could have any implementation?**

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

System.out.print("a");

**try**{

System.out.print("b");

e();

}

System.out.print("c");

}

**private** **static** **void** e() {

// code omitted

}

**○** A) a

**○** B) ab

**○** C) abc

**○** D) abc followed by a stack trace

**○** E) ab followed by a stack trace

**○** F) None of the above; the code does not compile as is.

1. **Which of the following are true about the enhanced for loop?(choose all that apply)**

**○** A) The terms must be compile-time constant values.

**○** B) Can be used to iterate over arrays

**○** C) Can be used to iterate over List objects

**○** D) The data type of the reference on the left-hand side must exactly match the data type of the elements on the right-hand side.

**○** E) Allows access to the built-in counter of the current loop position

1. **What is the result of the following code snippet?**

3: String[] values = { "one","two","three"};

4: **for**(**int** index = 0; index < values.length; index++)

5: System.***out***.print(values[index]);

6: System.***out***.print(index);

**○** A) onetwothree

**○** B) ontwothree2

**○** C) one0two1three2

**○** D) The code will not compile because of line 4.

**○** E) The code will not compile because of line 6.

1. **What is the result of the following code snippet?**

3: String year = "Senior";

4: **switch**(year) {

5: **case** "Freshman" : **default**: **case** "Sophomore":

6: **case** "Junior" : System.***out***.print("See you next year"); **break**;

7: **case** "Senior" : System.***out***.print("Congratulations");

8: }

**○** A) See you next year

**○** B) Congratulations

**○** C) See you next yearCongratulations

**○** D) The code does not output any text.

**○** E) The code will not compile because of line 4.

**○** F) The code will not compile because of line 5.

1. **What is the output of the following code?**

LocalDate date = LocalDate.*of*(2018, Month.***APRIL***, 30).plusMonths(-1).plusYears(20);

System.***out***.println(date.getYear()+" "+date.getMonthValue()+" "+date.getDayOfMonth());

**○** A) 2018 MARCH 30

**○** B) 2018 APRIL 30

**○** C) 2018 MAY 30

**○** D) 2038 MARCH 30

**○** E) 2038 APRIL 30

**○** F) 2038 MAY 30

1. **What is true of the following code?(Choose all that apply)**

1: **public** **class** Deer {

2: **private** **void** freeze() { *stayStill*(); }

3: **public** **static** **void** inTheHeadlights() { freeze(); }

4: **private** **static** **void** stayStill() {}

5: }

6: **public** **class** Car {

7: **public** **static** **void** drive(Deer deer) {

8: deer.*inTheHeadlights*();

9: } }

**○** A) Line 2 does not compile.

**○** B) Line 3 does not compile.

**○** C) Line 4 does not compile.

**○** D) Line 8 does not compile.

**○** E) The code compiles and runs without exception if Car.drive(null) is called.

**○** F) The code compiles but throws an exception if Car.drive(null) is called.

1. **What is the output of the following code?**

LocalDate d = LocalDate.*of*(2015,5);

Period p = Period.*of*(1,2,3);

d = d.minus(p);

DateTimeFormatter f = DateTimeFormatter.*ofLocalizedDateTime*(FormatStyle.***SHORT***);

System.***out***.println(f.format(d));

**○** A) 3/7/14 11:22 AM

**○** B) 6/7/14 11:22 AM

**○** C) 5/10/15 11:22 Am

**○** D) 6/10/15 11:22 AM

**○** E) The code does not compile.

**○** F) A runtime exception is thrown.

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

**○** A) For a String, capacity would be a redundant property.

**○** B) For a StringBuilder, capacity would be a redundant property.

**○** C) An empty String has a length of zero.

**○** D) An empty String has a length of one.

**○** E) A StringBuilder’s length is sometimes greater than its capacity.

**○** F) A StringBuilder’s length is never greater than its capacity.

1. **What does the following output?**

List numberList = Arrays.*asList*(5, 10, -5, -10);

Collections.*sort*(numberList);

**int** five = Collections.*binarySearch*(numberList,5);

**int** four = Collections.*binarySearch*(numberList,4);

System.***out***.println(five + four);

**○** A) -2

**○** B) -1

**○** C) 0

**○** D) 1

**○** E) 2

**○** F) An exception is thrown.

**○** G) The code doesn't compile.

1. **Which exception will the following throw?**

**int**[] nums =**new** **int**[] { 1, 0, 2};

Object p= nums;

**int**[] two = (**int**[]) p;

System.***out***.println(10 / two[2]);

**○** A) ArrayIndexOutOfBoundsException

**○** B) ClassCastException

**○** C) IllegalArgumentException

**○** D) NumberFormatException

**○** E) None of the above

1. **Which of the following is a reference variable (and not a primitive)?(Choose all that apply)**

**○** A) int[] ints;

**○** B) long[] longs;

**○** C) String[] strings;

**○** D) Object[] objects;

**○** E) None of the above

1. **Determine the output of the following code when executed with the command:**

java HelloWorld hello world goodbye

1: **public** **static** **class** HelloWorld {

2: **public** **static** **void** main(String[] args) {

3: System.***out***.println(args[1] + " " + args[2]);

4: } }

**○** A) hello world

**○** B) world goodbye

**○** C) null null

**○** D) An ArrayIndexOutOfBoundsOccurs at runtime.

**○** E) Does not compile

**○** F) Throws a different exception

1. **Which of the following methods compile?(Choose all that apply)**

**○** A) **boolean** isWaddling1(){

**return** true;

}

**○** B) **public** **boolean** isWaddling2(){

**return** 0;

}

**○** C) **boolean** isWaddling3(){

**return** **null**;

}

**○** D) **boolean** **private** isWaddling4(){

**return** **null**;

}

**○** E) **final** **boolean** isWaddling5(){

**return** **false**;

}

**○** F) **boolean** **final** isWaddling6(){

**return** **false**;

}

1. **What is the result of the following code snippet?**

3: **boolean** x = **false**;

4: **int** z = 0;

5: **do** {

6: **if**(z>5) {z++; x = **true**;}

7: **else** z += 3;

8: } **while** (!(x = **true**));

9: System.***out***.print(z);

**○** A) 8

**○** B) 0

**○** C) 7

**○** D) 3

**○** E) 6

**○** F) The code will not compile because of line 8.

**○** G) The code contains an infinite loop and does not terminate.

1. **What is the result of the following code snippet?**

3: String tiger = "Tiger";

4: String lion = "Lion";

5: **final** String statement = 250 > 380 ? lion : tiger = " is Bigger";

6: System.out.println(statement);

**○** A) Tiger

**○** B) Lion

**○** C) Tiger is Bigger

**○** D) Lion is Bigger

**○** E) is Bigger

**○** F) The code will not compile because of line 5.

1. **Which is the result of the following?**

List listOne = **new** ArrayList();

listOne.add(0.5); listOne.add(3.5);

List listTwo = **new** ArrayList();

listTwo.add(3.5); listTwo.add(0.5);

**if**(listOne == listTwo)

System.***out***.println("A");

**else** **if**(listOne.equals(listTwo))

System.***out***.println("B");

**else**

System.***out***.println("C");

**○** A) A

**○** B) B

**○** C) C

**○** D) An exception is thrown.

**○** E) The code does not compile.

1. **Which of braces can be removed without causing a compiler error?(Choose all that apply )**

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

**int** num1 = 8;

**int** num2 = 8;

**for**(**int** i=0; i < 3; i++) {

**if** (num1 == num2) {

**try** {

System.***out***.println("t");

}**catch** (Exception e) {

System.***out***.println("c");

}

}

}

}

**○** A) The braces for the method

**○** B) The braces for the loop

**○** C) The braces for the if statement

**○** D) The braces for the try

**○** E) The braces for the catch

**○** F) None of the above; the code does not compile as is.

1. **Which are true statements about the following code? (Choose all that apply)**

1: **public** **class** I {

2:

3: **int** ii;

4: **public** **void** i() {

5:

6: **for** (**int** i = 0; i< 5; i++) {

7:

8: }

9: } }

**○** A) int j=i; can be inserted at line 2.

**○** B) int j=i; can be inserted at line 5.

**○** C) int j=i; can be inserted at line 7.

**○** D) int j=ii; can be inserted at line 2.

**○** E) int j=ii; can be inserted at line 5.

**○** F) The code does not compile .

1. **What is the result of the following code ?**

List differentTypes = **new** ArrayList();

differentTypes.add("goldfish");

differentTypes.add(0, **false**);

differentTypes.add(1);

differentTypes.remove(1);

**boolean** b1 = differentTypes.contains("goldfish");

**boolean** b2 = differentTypes.contains(**new** Boolean(**false**));

**boolean** b3 = differentTypes.contains(1);

System.***out***.println(b1 + " " + b2 + " " + b3);

**○** A) false false true

**○** B) false true true

**○** C) true false false

**○** D) true true false

**○** E) An exception is thrown.

**○** F) The code does not compiles.

1. **What is result of this code?**

**public** **class** Cub {

**private** String name;

**private** **double** weight;

**public** Cub(**double** weight) {

**this**.weight = weight;

**this**("",weight);

}

**public** Cub(String name, **double** weight) {

weight = weight;

**this**.name = name;

}

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

Cub cub = **new** Cub(44);

System.***out***.println(cub.weight + "" + cub.name);

}

}

**○** A) 0

**○** B) 0 null

**○** C) 44

**○** D) 44 null

**○** E) 44.0

**○** F) 44.0 null

**○** G) The code does not compile.

1. **What is the result of the following code?**

**public** **class** Letters {

{ System.***out***.print("a"); }

**public** Letters() {

{ System.***out***.print("b"); }

}

{ System.***out***.print("c"); }

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

Letters a = **new** Letters();

{ System.***out***.print("d"); }

} }

**○** A) abcd

**○** B) acbd

**○** C) bacd

**○** D) bdac

**○** E) dbad

**○** F) The code does not compile.

1. **What is the result of the following code?(Choose all that apply)**

**class** Sparrow **extends** Bird { }

**public** **abstract** **class** Bird {

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

Bird b= **new** Sparrow();

Sparrow s = **new** Sparrow();

**if** (b == s) { System.***out***.println("=="); }

**if**( b.equals(s)) {

System.***out***.println("bird");

Sparrow s1 = **new** Sparrow();

}

**if** (b.equals(s1))

System.***out***.println("sparrow");

}

}

**○** A) Nothing is output.

**○** B) ==

**○** C) bird

**○** D) sparrows

**○** E) The code does not compile.

1. **What is the result of the following code snippet?**

**int** x = 4;

**if**(x <= 5 ^ x == 4)

System.***out***.print("Low");

**else** **if**(x == 4)

System.***out***.print("Match");

**if**(x > 1) System.***out***.print("High");

**○** A) Low

**○** B) Match

**○** C) High

**○** D) LowHigh

**○** E) MatchHigh

**○** F) The code will not compile because of line 4.

**○** G) The code will not compile because of line 8.

1. **Which methods will compile if inserted into Joey?(Choose all that apply)**

**class** CanNotHopException **extends** Exception {}

**class** Kangaroo {

**public** **void** hop() **throws** CanNotHopException { }

}

**class** Joey **extends** Kangaroo {

//INSERT CODE HERE

}

**○** A) public void hop() {}

**○** B) public void hop() throws Exception {}

**○** C) public void hop() throws CanNotHopException {}

**○** D) public void hop() throws RuntimeException {}

**○** E) None of these

1. **Given the following class definition, which method signature could appear in a subclass of Otter?(Choose all that apply)**

**public** **abstract** **class** Otter {

**protected** **abstract** **void** eatFish(**int** count);

}

**○** A) protected void eatFish(int count)

**○** B) protected void eatFish()

**○** C) protected int eatFish(int count)

**○** D) void eatFish(int fishCount)

**○** E) public void eatFish(int numberOfFish)

1. **Which methods will compile if inserted into Joey?(Choose all that apply)**

**class** Kangaroo {

**public** **void** hop() { }

}

**public** **class** Joey **extends** Kangaroo {

//INSERT CODE HERE

}

**○** A) public void hop() {}

**○** B) public void hop() throws Exception {}

**○** C) public void hop() throws IOException {}

**○** D) public void hop() throws RuntimeException {}

**○** E) None of these

1. **Which of the following fill in the blank to print out " Java"?(Choose all that apply)**

String letters = " Java ";

String answer = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

System.out.println(answer);

**○** A) letters.trim();

**○** B) letters.substring(1).trim();

**○** C) letters.trim().substring(1);

**○** D) letters.charAt(0) + letters.substring(1).toLowerCase();

**○** E) letters.replace(" ", "");

**○** F) letters.replace(" J", "");

1. **Which of the following lists is a correct sequence for catching exceptions? (Choose all that apply)**

**○** A) NumberFormatException, ClassCastException, RuntimeException

**○** B) NumberFormatException, RuntimeException, ClassCastException

**○** C) RuntimeException, NumberFormatException, ClassCastException

**○** D) RuntimeException, NumberFormatException, Exception

**○** E) NumberFormatException, Exception, RuntimeException

**○** F) NumberFormatException, RuntimeException, Exception

1. **What is the output when new Hippopotamus() is called?**

**class** Mammal {

**public** Mammal ( **int** age) { System.***out***.println("1"); }

}

**public** **class** Hippopotamus **extends** Mammal {

**public** Hippopotamus() { System.***out***.println("2"); }

}

**○** A) 12

**○** B) 1

**○** C) 2

**○** D) The code will not compile because of line 2.

**○** E) The code will not compile because of line 5.

**○** F) The code will compile but produce an error at runtime

1. **Which of the following are true about ternary operators?(Choose all that apply)**

**○** A) The left-hand side of the expression must evaluate to a boolean expression or numeric value of 0 or 1 at runtime.

**○** B) The data types of the two expressions on the right-hand side of the equation must match or be able to be numerically promoted to one another.

**○** C) All ternary operators can be rewritten as if-then-else statement.

**○** D) If the expression on the left-hand side evaluates to true, then the rightmost expression will not be evaluated.

**○** E) The terms must be compile-time constant values.

1. **Which are the minimum changes must be made to the following code for it to properly exhibit encapsulation?(Choose all that apply)**

**public** **class** HoneyPot {

Honey honey;

**public** Honey getHoney() { **return** honey; }

**public** **void** setHoney(Honey h) { honey = h; }

**public** **void** depositHoney(Bee bee) {

honey.add(bee.getHoney());

}

}

**○** A) Make the variable on line 2 private.

**○** B) Make the variable on line 2 public.

**○** C) Make the method on line 4 private.

**○** D) Make the method on line 5 private.

**○** E) Remove line 4.

**○** F) Remove lines 5-7.

1. **Which of the following compile? (Choose all that apply)**

**○** A) String[] grades; grades = {"B", "C", "F", "A", "D"};

**○** B) String[] grades; grades =**new** String[] {"B", "C", "F", "A", "D"};

**○** C) String[] grades = {"B", "C", "F", "A", "D"};

**○** D) String grades[] = {"B", "C", "F", "A", "D"};

**○** E) String grades[]; grades =**new** []String {"B", "C", "F", "A", "D"};

**○** F) String[] grades[] = {"B", "C", "F", "A", "D"};

1. **Which of the following exceptions are always thrown programmatically?(Choose all that apply)**

**○** A) ArrayIndexOutOfBounds

**○** B) ExceptionInInitializerError

**○** C) java.io.IOException()

**○** D) NullPointerException()

**○** E) NumberFormattException

1. **Which of the following identifiers are valid Java identifiers?(Choose allthat apply)**

**○** A) \_x

**○** B) if

**○** C) static

**○** D) com.my.company

**○** E) Abstract

**○** F) 2nd\_place

1. **What is the output of the code snippet?**

3: **int** j = 1;

4: **for**(**int** i = 0; i < 4 && j < 3; ++i){

5: ++j;

6: System.***out***.print(i+j);

7: }

**○** A) 0213

**○** B) 24

**○** C) 6

**○** D) 13

**○** E) The code will not compile because of line 4.

**○** F) The code contains an infinite loop and does not terminate.

1. **What is the output of the following code?**

**public** **class** HowMany {

**public** **static** **int** count(**int** a) {

**if** (a != 3) { **int** b = 1;

}**else** {

**int** b = 2;

}

**return** a++ +b;

}

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

System.***out***.println(*count*(3));

System.***out***.println(*count*(9));

} }

**○** A) 39

**○** B) 49

**○** C) 410

**○** D) 511

**○** E) The code fails to compile.

1. **What is the result of the following code?(Choose all that apply)**

**public** **class** Cardinal {

**static** **int** *number*;

Cardinal() { *number*++; }

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

Cardinal c1 = **new** Cardinal();

**if** (c1 == **null**) {

Cardinal c2 = **new** Cardinal();

}

Cardinal c2 = **new** Cardinal();

System.***out***.println(c1.*number*);

}

**package** bird;

}

**○** A) 0

**○** B) 1

**○** C) 2

**○** D) 3

**○** E) 4

**○** F) The code does not compile.

1. **Given this array declaration, which of the following statements are valid array indexes?**

int[][] numbers = {{1,2}, {3}, {4,5,6}};

**○** A) numbers[0][1]

**○** B) numbers[0][2]

**○** C) numbers[1][0]

**○** D) numbers[1][1]

**○** E) numbers[2][0]

**○** F) numbers[3][1]

1. **The following code appears in a file named Plant.java. What is the result of compiling this source file?**

**public** **class** Plant {

**public** **boolean** flowering;

**public** Leaf [] leaves;

}

**class** Leaf {

**public** String color;

**public** **int** length; }

**○** A) The code compiles successfully and two bytecode files are generated:Plant.class and Leaf.class.

**○** B) The code code compiles successfully and one bytecode file is generated:Plant.class.

**○** C) A compiler error occurs on line 1.

**○** D) A compiler error occurs on line 3.

**○** E) A compiler error occurs on line 5.

**○** F) A compiler error occurs on another line.

1. **What is the result of the following?**

String[] lizards = {"iguana", "gecko"};

List<String> list = Arrays.*asList*(lizards);

list.set(1, "");

System.***out***.println(list);

**○** A) [iguana]

**○** B) [iguana, ]

**○** C) [iguana, gecko]

**○** D) An exception is thrown on line 8.

**○** E) An exception is thrown on line 9.

**○** F) The code does not compile.

**85. What is the result of the following code?(Choose all that apply)**

**public** **class** Cardinal {

**static** **int** *number*;

Cardinal() { *number*++; }

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

Cardinal c1 = **new** Cardinal();

**if** (c1 == **null**) {

Cardinal c2 = **new** Cardinal();

}**else** {

Cardinal c2 = **new** Cardinal();

}

Cardinal c2 = **new** Cardinal();

System.***out***.println(c1.*number*);

}

}

**○** A) 0

**○** B) 1

**○** C) 2

**○** D) 3

**○** E) 4

**○** F) The code does not compile.

1. **What is the output when new PrayingMantis("green") is called?**

1: **abstract** **class** Insect {

2: **public** Insect (**int** age) { System.***out***.println("1"); }

3: **public** Insect(String color) { **this**(5); System.***out***.println("2"); }

4: }

5: **public** **class** PrayingMantis **extends** Insect {

6: **public** PrayingMantis( String color) {

7: System.***out***.println("3");

8: }

9: }

**○** A) 123

**○** B) 13

**○** C) The code will not compile because of line 1.

**○** D) The code will not compile because of line 3.

**○** E) The code will not compile because of line 7.

**○** F) The code compiles but throws an exception at runtime.

1. **Which are true statements about the following code?(Choose all that apply)**

**public** **class** Type {

**int** integer;

**static** **int** *num*;

**public** Type() {

**int** type;

} }

**○** A) The code compiles.

**○** B) The code does not compile.

**○** C) Exactly one variable is automatically initialized.

**○** D) Exactly two variable are automatically initialized.

**○** E) num is a class variable.

**○** F) integer is a class variable.

1. **What is the result of the following class?**

1: **import** java.util.function.\*;

2: **public** **class** Panda {

3: **int** age;

4: **public** **static** **void** main(String[] args) {

5: Panda p1 = **new** Panda();

6: p1.age = 1;

7: check(p1, Panda p -> p.age < 5);

8: }

9: **private** **static** **void** check(Panda panda, Predicate<Panda>pred) {

10: String result = pred.test(panda) ? "match" : "not match";

11: System.***out***.println(result);

12: } }

**○** A) match

**○** B) not match

**○** C) Compiler error on line 8

**○** D) Compiler error on line 10

**○** E) Compiler error on line 11

**○** F) A runtime exception is thrown.

1. **Which of the following are included as a result of this code?**

String s = "Hello";

String t = **new** String(s);

**if**("Hello".equal(s)) System.***out***.println("one");

**if**(t == s) System.***out***.println("two");

**if**(t.equal(s)) System.***out***.println("three");

**if**("Hello" == s) System.***out***.println("four");

**if**("Hello" == t) System.***out***.println("five");

**○** A) one

**○** B) two

**○** C) three

**○** D) four

**○** E) five

**○** F) The code does not compile.

1. **What is the output of the following program?**

**public** **class** Pet {

**public** **void** eat() **throws** NullPointerException {

**throw** **new** NullPointerexception();

}

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

Pet pet = **new** Pet();

pet.eat();

System.***out***.println("Pet just ate");

} }

**○** A) Pet just ate

**○** B) Line 2 generates a compiler error.

**○** C) Line 3 generates a compiler error.

**○** D) Line 7 generates a compiler error.

**○** E) A NullPointerException is thrown at runtime.

1. **Which of the following compile?(Choose all that apply)**

**○** A) **public** **void** swim(**int**... distance){}

**○** B) **public** **void** swim(**int**... distance, **int** time){}

**○** C) **public** **void** swim(**int** time, **int**... distance){}

**○** D) **public** **void** swim(**int**... distance, **int**,,, time){}

**○** E) **public** **void** swim(**int** time, ...**int** distance){}

1. **What is the result of the following code?**

1: **public** **class** Counts {

2: **private** **boolean** b;

3: **public** **static** **void** main(String[] args) {

4: Counts c = **new** Counts();

5: **int** one, two = 0;

6: **if**(c.b) {

7: System.***out***.println(one);

8: } } }

**○** A) Compiler error on line 5

**○** B) Compiler error on line 6

**○** C) Compiler error on line 7

**○** D) Compiles successfully and prints 0

**○** E) Compiles successfully and prints null

**○** F) Compiles successfully with no output

1. **Which of the following are true of the following code? (Choose all that apply)**

**public** **class** Giraffe {

**private** **int** height;

**private** **int** age;

**boolean** tired;

**public** **boolean** isTired() { **return** tired; }

**public** **int** getHeight() { **return** height; }

**public** **void** setHeight(**int** h) { **this**.height = h; }

}

**○** A) The data in this class is encapsulated.

**○** B) This class currently is immutable.

**○** C) Making line 4 private would exhibit data encapsulation.

**○** D) Making line 4 private would exhibit immutability.

**○** E) The method names follow JavaBean conventions.

1. **What is the result of the following code snippet?**

**int** x = 9;

**long** y = x\* (**long**) (++x);

System.***out***.println(y);

**○** A) -1

**○** B) 9

**○** C) 81

**○** D) 90

**○** E) The code will not compile because of line 4.

1. **What is the result of this code?(Choose all that apply)**

**public** **class** Change {

**public** **static** **void** change(**int** size, StringBuilder s) {

s.append("b");

size++;

}

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

**int** size = 2;

StringBuilder s1 = **new** StringBuilder("s1");

*change*(size, s1);

System.***out***.println("s1 = " + s1);

System.***out***.println("size = "+size);

}

}

**○** A) s1 = s1

**○** B) s1 = s1b

**○** C) size = 2

**○** D) size = 3

**○** E) The code does not compile.

**○** F) An exception is thrown.

1. **Which of these lines compile?(Choose all that apply)**

1: **public** **class** Rich {

2: **public** **void** money() {

3: **int** \_million = 1\_000\_000;

4: **double** aThousand = 1\_000\_.00;

5: **double** 100 = 100;

6: **int** hundred = 100.00;

7: **float** ten = 10d;

8: **short** one = 1;

9: }

10: }

**○** A) 3

**○** B) 4

**○** C) 5

**○** D) 6

**○** E) 7

**○** F) 8

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

**○** A) Java allows extending from multiple classes directly.

**○** B) Java allows implementing multiple interfaces directly.

**○** C) Java code compiled on Unix must be recompile to run on Windows.

**○** D) Java supports method overloading.

**○** E) Java supports operator overloading.

1. **Which statements are true? (Choose all that apply)**

**○** A) If a try block contains System.exit(0), the finally block will run.

**○** B) If a try block contains System.exit(0), the finally block will not run.

**○** C) If a catch block contains System.exit(0), the finally block will run.

**○** D) If a catch block contains System.exit(0), the finally block will not run.

**○** E) A try block will not compile containing System.exit(0).

1. **Which are true the following code?**

**public** **class** C {

String seq = "c";

**static** { seq += "g"; }

{ seq += "z"; }

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

C c1 = **new** C();

C c2 = **new** C();

System.***out***.println(c1.seq);

}

}

**○** A) The code compiles and outputs gzz.

**○** B) The code compiles and outputs gzgz.

**○** C) If the lines that do not compile are removed, the code outputs cg.

**○** D) If the lines that do not compile are removed, the code outputs cz.

**○** E) If the lines that do not compile are removed, the code outputs cgzz.

**○** F) If the lines that do not compile are removed, the code outputs cgzg.

1. **Suppose we have the following classes named InventoryItem and Order. Which one of the following statements inserted at line 2 of the Order class will make the Order class compile successfully?**

1: **package** com.abc.products;

2: **public** **class** InventoryItem { }

1: **package** com.abc.orders;

2: //INSERT CODE HERE

3: **public** **class** Order {

4: InventoryItem [] items;

5: }

**○** A) import com.abc.products;

**○** B) import com.abc.products.\*;

**○** C) import static com.abc.products

**○** D) import com.abc.products.InventoryItem.\*

**○** E) Nothing; the class compiles as is.

1. **What is the output of the following code?**

LocalDateTime d = LocalDateTime.*of*(2015, 5, 10, 11, 22, 33);

DateTimeFormatter f = DateTimeFormatter.*ofPattern*("hh:MM");

System.***out***.println(d.format(f));

**○** A) 11:05

**○** B) 11:22

**○** C) 22:05

**○** D) 22:33

**○** E) The code does not compile.

**○** F) A runtime exception is thrown.

1. **Which of the following correctly overload this method? (Choose all that apply)**

**public** **void** buzz() { }

**○** A) **private** **void** buzz(String sound) { }

**○** B) **public final** **void** buzz() { }

**○** C) **public static** **void** buzz() { }

**○** D) **public static** **void** buzz(int… time) { }

**○** E) **public** **void** buzz(Boolean softly) { }

**○** F) **public** **void** buzzLouder() { }

1. **Which are true of the following code? (Choose all that apply)**

1: **package** aquarium;

2: **public** **class** Water {

3: **public** String toString() { **return** ""; } }

1: **package** aquarium;

2: **public** **class** Shark {

3: **static** **int** *numFins*;

4: **static** Water *water*;

5: **public** **static** **void** main(String[] args) {

6: String s1 = *water*.toString();

7: String s2 = *numFins*.toString();

8: }

9: }

**○** A) Shark.java needs an additional import to compile.

**○** B) water.toString() does not compile.

**○** C) numFins.toString does not compile.

**○** D) Water.java does not compile.

**○** E) java Shark will run the program and produce no output.

**○** F) The code compiles as is.

1. **What is the output of the following code?**

LocalDateTime d = LocalDateTime.*of*(2015, 5, 10, 11, 22, 33);

Period p = Period.*of*(1, 2, 3);

d.minus(p);

DateTimeFormatter f = DateTimeFormatter.*ofLocalizedDateTime*(FormatStyle.***SHORT***);

System.***out***.println(f.format(d));

**○** A) 3/7/14 11:22 AM

**○** B) 6/7/14 11:22 AM

**○** C) 5/10/15 11:22 Am

**○** D) 6/10/15 11:22 AM

**○** E) The code does not compile.

**○** F) A runtime exception is thrown.

1. **What is the result of this code?**

**public** **class** HowMany {

**public** **int** add(**int** one, **int** two){

**return** one + two;

}

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

HowMany h = **new** HowMany();

**int** result = h.add(2, 3);

System.***out***.println(result);

}

}

**○** A) 2

**○** B) 3

**○** C) 5

**○** D) The code does not compile.

**○** E) An exception is thrown.

1. **What is the output of the following program?**

**public** **class** Image {

**public** **int** width, height;

**public** **void** showImage() **throws** Exception {

**if**(width < 0 || height < 0)

**throw** **new** Exception("invalid image size");

**else**

System.***out***.print("1");

}

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

Image image = **new** Image();

image.width = -10;

**try**{

image.showImage();

System.***out***.print("2");

}**catch**(Exception e){

System.***out***.print("3");

}

System.***out***.print("4");

}

}

**○** A) 124

**○** B) 234

**○** C) 34

**○** D) 3 and a stack trace for Exception

**○** E) 34 and a stack trace for Exception

**○** F) The code does not compile.

1. **Suppose we have the following class named ShowDate. Which one of the following statements is true?**

**import** java.util.Date;

**public** **class** ShowDate {

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

Date a = **new** Date();

Date b = **new** Date();

Date c = a;

System.***out***.println(c.toString());

a = **null**;

c = **null**;

}

}

**○** A) The Date object from line 4 is eligible for garbage collection immediately following line 6.

**○** B) The Date object from line 4 is eligible for garbage collection immediately following line 8.

**○** C) The Date object from line 4 is eligible for garbage collection immediately following line 9.

**○** D) The code does not compile.

**○** E) The code throws an exception.

1. **What is the result of compiling the following code?(Choose all that apply)**

1: **abstract** **class** Mammal {

2: **public** **abstract** **int** getAge();

3: }

4: **abstract** **class** Animal {

5: **public** **int** getAge();

6: }

7: **abstract** **interface** Herbivore {}

8: **public** **class** HoneyBadger **extends** Mammal, Animal **implements** Herbivore {

9: **int** getAge(){**return** 5; }

10: }

**○** A) The code compiles without issue.

**○** B) The code will not compile because of line 2.

**○** C) The code will not compile because of line 5.

**○** D) The code will not compile because of line 8.

**○** E) The code will not compile because of line 9.

**○** F) The code compiles but throws an exception at runtime.

1. **Given the following code, which statements are true?(Choose all that apply)**

**import** **static** java.lang.System.***out***;

**class** Flower { }

**public** **class** Bee {

**void** pollinate(Flower flower) { ***out***.println(1); }

**void** pollinate(Flower... flower) { ***out***.println(2); }

**void** pollinate(**int**... numFlowers) { ***out***.println(3); }

**void** pollinate(Integer numFlowers) { ***out***.println(4); }

**void** pollinate(String s) { ***out***.println(5); }

**void** pollinate(Object obj) { ***out***.println(6); }

}

**○** A) Calling pollinate(new Flower())prints 1.

**○** B) Calling pollinate(new Flower())prints 2.

**○** C) Calling pollinate(1)prints 3.

**○** D) Calling pollinate(1)prints 4.

**○** E) Calling pollinate("flower")prints 5.

**○** F) Calling pollinate("flower")prints 6.

1. **What gets printed when running the following?(Choose all that apply)**

**public** **class** A {

**private** String value;

**public** **void** go() {

**try** {

System.***out***.print(value.toString());

System.***out***.print("1");

}**catch** (NullPointerException e) {

System.***out***.print("2");

System.*exit*(0);

}**finally** {

System.***out***.print("3");

}

System.***out***.print("4");

}

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

**new** A().go();

}

}

**○** A) 1

**○** B) 2

**○** C) 3

**○** D) 4

**○** E) Stack trace of a NullPointerException

1. **What is the output of the following code?**

**public** **class** BigCat {

**protected** **final** **void** purr() { System.***out***.println("BigCat purrs!"); }

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

BigCat bigCat = **new** Ocelot();

bigCat.purr();

}

}

**class** Ocelot **extends** BigCat {

**protected** **final** **void** purr() { System.***out***.println("Ocelot purrs!"); }

}

**○** A) BigCat purrs!

**○** B) Ocelot purrs!

**○** C) The code will not compile because of line 4.

**○** D) The code will not compile because of line 5.

**○** E) The code will not compile because of line 9.

1. **What is the result of following code?**

**public** **class** A {

**private** **int** i = 6;

**private** **int** j = i;

**public void** A() {

i = 5;

}

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

A a = **new** A();

System.***out***.println(a.i + a.j);

}

}

**○** A) 55

**○** B) 56

**○** C) 66

**○** D) 10

**○** E) 11

**○** F) 12

**○** G) The code does not compile.

1. **The following code appears in a file named Flight.java. What is the result of compiling this source file?**

1: **public** **class** Flight {

2: **private** FlightNumber number;

3:

4: **public** Flight(FlightNumber number){

5: **this**.number = number;

6: } }

7: **public** **class** FlightNumber {

8: **public** **int** value;

9: **public** String code; }

**○** A) The code compiles successfully and two bytecode files are generated: Flight.class and FlightNumber.class.

**○** B) The code compiles successfully and one bytecode file is generated: Flight.class.

**○** C) A compiler error occurs on line 2.

**○** D) A compiler error occurs on line 4.

**○** E) A compiler error occurs on line 7.

1. **What is the result of the following class?**

1: **import** java.util.function.\*;

2:

3: **public** **class** Panda {

4: **int** age;

5: **public** **static** **void** main(String[] args) {

6: Panda p1 = **new** Panda();

7: p1.age = 1;

8: *check*(p1, p -> {p.age < 5});

9: }

10: **private** **static** **void** check(Panda panda, Predicate<Panda>pred) {

11: String result = pred.test(panda) ? "match" : "not match";

12: System.***out***.println(result);

13: } }

**○** A) match

**○** B) not match

**○** C) Compiler error on line 8

**○** D) Compiler error on line 10

**○** E) Compiler error on line 11

**○** F) A runtime exception is thrown.

1. **What is the output of the following snippet?**

Integer x = **null**;

**int** y = 0;

**if**( x == y){

System.***out***.print("1");

}**else** {

System.***out***.print("2");

} **finally** {

System.out.print("3");

}

**○** A) 13

**○** B) 23

**○** C) 2 and the stack trace for a NullPointerException

**○** D) 23 and the stack trace for a NullPointerException

**○** E) 234

**○** F) The code does not compile.

1. **Which lines in Tadpole give compiler error? (Choose all that apply)**

1: **package** animal;

2: **public** **class** Frog {

3: **protected** **void** ribbit() { }

4: **void** jump() { }

5: }

1: **package** other;

2: **import** animal.\*;

3: **public** **class** Tadpole **extends** Frog {

4: **public** **static** **void** main(String[] args) {

5: Tadpole t = **new** Tadpole();

6: t.ribbit();

7: t.jump();

8: Frog f = **new** Tadpole();

9: f.ribbit();

10: f.jump();

11: }

12: }

**○** A) 5

**○** B) 6

**○** C) 7

**○** D) 8

**○** E) 9

**○** F) 10

1. **What is the result of compiling the following code?**

1: **interface** CanSwim {

2: **public** **static** **int** ***SPEED***;

3: **public** **void** swim();

4: }

5: **public** **class** MantaRay **implements** CanSwim {

6: **public** **void** swim() { System.***out***.println("MantaRay is swimming: "+***SPEED***); 7: }

8: }

**○** A) The code compiles without issue.

**○** B) The code will not compile because of line 2.

**○** C) The code will not compile because of line 3.

**○** D) The code will not compile because of line 5.

**○** E) The code will not compile because of line 6.

**○** F) The code compiles but throws an exception at runtime.

1. **Which of the following statements can be inserted in the blank so that the code will compile successfully?(Choose all that apply)**

**public** **interface** WalksOn4Legs {}

**public** **abstract** **class** Mammal {

**public** **int** numberOfOffspring;

}

**public** **class** Antelope **extends** Mammal **implements** WalksOn4Legs {}

**public** **class** ParkRanger {

**public** **void** noteNewOffspring(Mammal mammal) { mammal.numberOfOffspring++; }

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

**new** ParkRanger().noteNewOffspring(\_\_\_\_\_\_\_\_\_\_\_\_);

}

}

**○** A) new Mammal()

**○** B) new Antelope()

**○** C) new WalksOn4Legs()

**○** D) (Mammal)new Object()

**○** E) (Mammal)new String()

**○** F) null

1. **Given the following class definitions, which method signature could appear in a subclass of Albatross? (Choose all that apply)**

**public** **abstract** **class** SeaBird {

**public** **abstract** **void** fly(**int** height);

}

**public** **abstract** **class** Albatross {

**abstract** Long fly();}

**○** A) private abstract void fly(int height)

**○** B) protected void fly(int height)

**○** C) public Number fly()

**○** D) Long fly()

**○** E) protected Long fly()

1. **Which of the following fill in the blank to remove the blank space in this String?(Choose all that apply)**

String g = "Guinea Pig";

**int** i = g.indexOf(" ");

String answer = \_\_\_\_\_\_\_\_\_

System.out.println(answer);

**○** A) g.trim();

**○** B) g.substring(0, i) + g.substring(i);

**○** C) g.substring(0, i - 1) + g.substring(i);

**○** D) g.substring(0, i) + g.substring(i + 1);

**○** E) g.substring(0, i - 1) + g.substring(i + 1);

**○** F) None of the above

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

StringBuilder s1 = **new** StringBuilder("meow");

StringBuilder s2 = **new** StringBuilder("meow");

**if**( s1 == s2) System.***out***.println("one");

**if**( s1.equals(s2)) System.***out***.println("two");

**if**( s1 == "meow") System.***out***.println("three");

**if**( s1.toString() == "meow") System.***out***.println("four");

**○** A) The code compiles.

**○** B) The code does not compile.

**○** C) If statements that do not compile are removed, the output contains one.

**○** D) If statements that do not compile are removed, the output contains two.

**○** E) If statements that do not compile are removed, the output contains three.

**○** F) If statements that do not compile are removed, the output contains four.

1. **What is the result of the following code?**

1: **public** **class** PrintCharecters {

2: **public** **static** **void** main(String[] args) {

3: **char** value = 'c';

4: **do** System.***out***.print(value++);

5: **while** (value <= 'f');

6: } }

**○** A) cde

**○** B) cdef

**○** C) def

**○** D) defg

**○** E) The code will not compile because of line 4.

**○** F) The code will not compile because of line 5.

1. **What is the result of this code?**

**class** Toy{

**private** **boolean** containsIce = **false**;

**public** **boolean** containsIce() {

**return** containsIce;

}

**public** **void** removeIce() {

**this**.containsIce = **true**;

}

}

**public** **class** Otter {

**private** **static** **void** play(Toy toy) {

toy.removeIce();

}

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

Toy toy = **new** Toy();

Otter.*play*(toy);

System.***out***.println(toy.containsIce());

}

}

**○** A) false

**○** B) true

**○** C) There is one compiler error.

**○** D) There are two compiler error.

**○** E) There are three compiler error.

**○** F) An exception is thrown.

1. **Which methods will compile if inserted into Joey?(Choose all that apply)**

**class** CanNotHopException **extends** Exception {}

**class** Kangaroo {

**public** **void** hop() **throws** CanNotHopException { }

}

**class** Joey **extends** Kangaroo {

//INSERT CODE HERE

}

**○** A) public void hop() {}

**○** B) public void hop() throws Exception {}

**○** C) public void hop() throws CanNotHopException {}

**○** D) public void hop() throws RuntimeException {}

**○** E) None of these

1. **What is printed when running the following snippet?**

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

**int** i = 0;

**try** {

i += 1;

*e*();

i += 2;

} **catch** (Exception e) {

i += 4;

}

System.***out***.print(i);

}

**private** **static** **void** e() {

**throws** **new** IllegalArgumentException();

}

**○** A) 1

**○** B) 3

**○** C) 5

**○** D) 7

**○** E) The code does not compile.

**○** F) An exception is thrown.

1. **What is the result of the following code snippet?**

3: **int** x = 10;

4: **switch**(x % 4.) {

5: **default**: System.***out***.print("Not divisible by 4");

6: **case** 0: System.***out***.print("Divisible by 4");

7: }

**○** A) Not divisible by 4

**○** B) Divisible by 4

**○** C) Not divisible by 4Divisible by 4

**○** D) The code does not output any text.

**○** E) The code does not compile because of line 4.

1. **Given the following class definition, which method signature could appear in a subclass of Otter ? (Choose all that apply)**

**public** **abstract** **class** Otter {

**protected** **abstract** **void** eatFish(**int** count);

}

**○** A) protected void eatFish(int count)

**○** B) protected void eatFish()

**○** C) protected int eatFish(int count)

**○** D) void eatFish(int fishCount)

**○** E) public void eatFish(int numberOfFish)

1. **What is the result of this code? (Choose all that apply)**

1: **public** **class** Dino {

2: **static** **final** String ***species***;

3: **double** weight;

4: { ***species*** = "Raptor"; }

5: **public** Dino(**double** weight){

6: **this**.weight = weight;

7: }

8: **public** **static** **void** main(String[] args) {

9: Dino dino = **new** Dino(500);

10: System.***out***.println(dino.weight);

11: } }

**○** A) Compiler error on line 2

**○** B) Compiler error on line 4

**○** C) Compiler error on line 9

**○** D) Compiler error on line 10

**○** E) 500

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

StringBuilder b = **new** StringBuilder("1");

StringBuilder c = b.append("2");

b.append("234");

c.deleteCharAt(1);

System.***out***.println("Equals?" + (b == c));

System.***out***.println("B=" + b);

System.***out***.println("C=" + c);

**○** A) The output contains Equals? true.

**○** B) The output contains Equals? false.

**○** C) The output contains B=1234.

**○** D) The output contains B=234.

**○** E) The output contains C=1234.

**○** F) The output contains C=2.

**○** G) The code does not compile.

1. **What is the result of the following code assuming garbage collection runs on line 6?**

**public** **class** Caterpillar {

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

Caterpillar c1 = **new** Caterpillar();

Caterpillar c2 = **new** Caterpillar();

c1 = c2; c2 = **null**;

// garbage collection runs

c1 = **null**;

}

**protected** **void** finalize() {

System.***out***.println("becames a butterfly");

} }

**○** A) "becomes a butterfly" is never printed.

**○** B) "becomes a butterfly" is printed exactly once.

**○** C) "becomes a butterfly" is printed exactly twice.

**○** D) The code fails compilation on line 3.

**○** E) The code fails compilation on line 9.

**○** F) The code fails compilation on another line.

1. **What is the output of the following code?**

1: **interface** CanClimb { **int** maxHeight(); }

2: **interface** HasClaws { **boolean** isSharp(); }

3: **public** **class** Koala **implements** CanClimb, HasClaws{

4: **public** **boolean** isSharp() { **return** **true**; }

5: **public** **int** maxHeight() { **return** 15; }

6: **public** **static** **void** main(String[] args) {

7: Object koala = **new** Koala();

8: CanClimb canClimb = (CanClimb)koala;

9: HasClaws hasClaws = (HasClaws)canClimb;

10: System.***out***.print(canClimb.maxHeight());

11: System.***out***.print(hasClaws.isSharp());

12: }

13: }

**○** A) true15

**○** B) 15true

**○** C) The code will not compile because of line 7.

**○** D) The code will not compile because of line 8.

**○** E) The code will not compile because of line 9.

**○** F) The code compiles but throws an exception at runtime.

1. **What is the result of the following code?**

**public** **class** MessageLogger {

**public** **static** **void** log(String s) {

System.***out***.print("s");

}

**public** **static** **void** log(**int** i) {

System.***out***.print("i");

}

**public** **static** **void** log(**double** d) {

System.***out***.print("d");

}

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

**short** s = 123;

*log*(s);

*log*(25.0);

*log*("hello");

}

}

**○** A) dds

**○** B) dis

**○** C) ids

**○** D) iis

**○** E) The code does not compile.

1. **When does the string object instantiated on line 3 become eligible for garbage collection?**

1: **public** **class** ReturnDemo {

2: **public** **static** String getName() {

3: String temp = **new** String("Jane Doe");

4: **return** temp;

5: }

6: **public** **static** **void** main(String[] args) {

7: String result;

8: result = *getName*();

9: System.***out***.println(result);

10: result = **null**;

11: System.*gc*();

12: }

13: }

**○** A) Immediately after line 4.

**○** B) Immediately after line 5.

**○** C) Immediately after line 9.

**○** D) Immediately after line 10.

**○** E) Immediately after line 11.

**○** F) Immediately after line 12.

**○** G) The code does not compile.

1. **What is the result of the following code ?**

1: **public** **class** ModSample {

2: **public** **static** **void** main(String[] args) {

3: **int** y = 4;

4: **int** x = 10 - ++y / 5;

5: System.***out***.println(x % y);

6: }

7: }

**○** A) 4

**○** B) 0

**○** C) 7

**○** D) 5

**○** E) The answer cannot be determined until runtime.

**○** F) The code will not compile because of line 4.

1. **Which of the following are supported by Java?(Choose all that apply)**

**○** A) Encapsulation

**○** B) Indentation changing the meaning of code

**○** C) Inheritance

**○** D) Operator overloading

**○** E) Platform independence

**○** F) Pointers

1. **What is the output of the following program?**

**public** **class** ColorPicker {

**public** **void** pickColor() {

**try** {

System.***out***.print("A");

fail();

} **catch** (NullPointerException e) {

System.***out***.print("B");

} **finally** {

System.***out***.print("C");

}

}

**public** **void** fail() {

**throw** **new** ArithmeticException();

}

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

**new** ColorPicker().pickColor();

System.***out***.print("D");

} }

**○** A) ABCD

**○** B) ABD

**○** C) A and a stack trace for ArithmeticException

**○** D) AC and a stack trace for ArithmeticException

**○** E) ACD and a stack trace for ArithmeticException

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

**○** A) Garbage collection runs when System.gc() is called.

**○** B) Finalize() is called if an object is garbage collected.

**○** C) finalize() is always called on an object.

**○** D) finalize() can be called twice on the same object.

**○** E) finalize() can be called twice for the same class.

1. **What is the result of the following code?**

1: **interface** Climb {

2: **boolean** isTooHigh( **int** height, **int** limit);

3: }

4:

5: **public** **class** Climber {

6: **public** **static** **void** main(String[] args) {

7: *check*((h,l) -> h > l,5);

8: }

9: **private** **static** **void** check(Climb climb, **int** height ){

10: **if** (climb.isTooHigh(height, 10))

11: System.***out***.println("too high");

12: **else**

13: System.***out***.println("ok");

14: }

15: }

**○** A) ok

**○** B) too high

**○** C) Compiler error on line 7

**○** D) Compiler error on line 10

**○** E) Compiler error on a different line

**○** F) A runtime exception is thrown.

1. **How many of the following lines give a compiler error?**

**public** **class** \_C {

String s = **null**;

**static** **int** 123;

**void** **char** = 'a';

**byte** b1 = 2;

**int** c1$ = b1;

**short** s1 = c1$;

**long** e\_2 = c1$;

}

**○** A) 1

**○** B) 2

**○** C) 3

**○** D) 4

**○** E) 5

1. **What is the output of the following code?**

1: **class** Animal {

2: **public** **int** getAge() {**return** 10;}

3: }

4: **class** Mammal **extends** Animal {

5: **protected** **int** getAge(**int** input) {**return** 7;}

6: }

7: **public** **class** Sloth **extends** Mammal {

8: **public** **boolean** hasFur() {**return** **true**;}

9: **public** **static** **void** main(String[] args) {

10: Mammal sloth = **new** Sloth();

11: System.***out***.print(sloth.getAge());

12: System.***out***.print(sloth.getAge(2));

13: System.***out***.print(sloth.hasFur());

14: }

15: }

**○** A) 7true10

**○** B) 10true7

**○** C) The code will not compile because of line 5.

**○** D) The code will not compile because of line 11.

**○** E) The code will not compile because of line 12.

**○** F) The code will not compile because of line 13.

1. **Which is the result of the following?**

List<String> numbers = **new** ArrayList<>();

numbers.add("4"); numbers.add("7");

numbers.set(1, "5");

numbers.add("8");

numbers.remove(0);

**for** (String number : numbers) System.***out***.print(number);

**○** A) 48

**○** B) 58

**○** C) 478

**○** D) 578

**○** E) 78

**○** F) An exception is thrown.

**○** G) The code does not compile.

1. **What is the result of the following code?**

**public** **class** Lion {

Lion l = **new** Lion();

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

**new** Lion();

}

**public** **void** roar() {

Lion l = **new** Lion();

**if**( l == l) {

System.***out***.println("roar!");

} } }

**○** A) Outputs nothing

**○** B) Outputs roar!

**○** C) The program never terminates.

**○** D) The code does not compile.

**○** E) The code throws an exception.

1. **What is the output of the following program?**

**public** **class** Supper {

**public** **static** **void** eat() **throws** IOException {

**try** {

System.***out***.print("1");

**throw** **new** IOException();

}**catch**(IOException e) {

System.***out***.println("2");

**throw** e;

}**finally** {

System.***out***.println("3");

}

}

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

*eat*();

System.***out***.println("4");

}

}

**○** A) 1234

**○** B) 12 and a stack trace for IOException

**○** C) 123 and a stack trace for IOException

**○** D) 1234 and a stack trace for IOException

**○** E) The code does not compile.

1. **Which of the following can fill in the blank on line 5 to make the code compile?(Choose all that apply)**

1: **package** x.y;

2: **public** **class** Movie {

3: **private** String title;

4: **protected** String rating;

5: **int** length;

6: **public** String description;

7: }

1: **package** x.y;

2: **public** **class** MovieTheater {

3: **private** Movie movie = **new** Movie();

4: **public** MovieTheater() {

5: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6: }

7: }

**○** A) movie.title = "Office Space";

**○** B) movie.rating = "R";

**○** C) movie.length = 89;

**○** D) movie.description = "Comedy";

**○** E) None of the above

1. **What is the result of the following code?**

1: **public** **class** BytePrinter {

2: **public** **void** print(**byte** b) {

3: System.***out***.print("byte");

4: }

5: **public** **void** print(Byte b) {

6: System.***out***.print("Byte");

7: }

8: **public** **void** print(**int** i) {

9: System.***out***.print("int");

10: }

11: **public** **static** **void** main(String[] args) {

12: BytePrinter printer = **new** BytePrinter();

13: **short** x = 10;

14: **byte** y = 12;

15: printer.print(x);

16: printer.print(y);

17: }

18: }

**○** A) byteint

**○** B) intbyte

**○** C) Compiler error on line 5

**○** D) Compiler error on line 15

**○** E) Compiler error on line 16

**146. What is the result of compiling this class?**

1: **public** **class** Frog {

2: **public** **int** Frog() { **return** 0; }

3: **private** List<Integer> legs;

4: **public** Frog() {

5: legs = **new** ArrayList<Integer>();

6: **for**( **int** i = 0; i<4; i++) {

7: legs.add(i);

8: } } }

**○** A) The code compiles successfully.

**○** B) The first compiler error occurs on line 2.

**○** C) The first compiler error occurs on line 3.

**○** D) The first compiler error occurs on line 4.

**○** E) The first compiler error occurs on line 5.

**○** F) The first compiler error occurs on line 6.

**○** G) The first compiler error occurs on line 7.