**Object Oriented Concepts**

## Questions

1. Real-world objects contain \_\_\_ and \_\_\_.
2. A software object's state is stored in \_\_\_.
3. A software object's behavior is exposed through \_\_\_.
4. Hiding internal data from the outside world, and accessing it only through publicly exposed methods is known as data \_\_\_.
5. A blueprint for a software object is called a \_\_\_.
6. Common behavior can be defined in a \_\_\_ and inherited into a \_\_\_ using the \_\_\_ keyword.
7. A collection of methods with no implementation is called an \_\_\_.
8. A namespace that organizes classes and interfaces by functionality is called a \_\_\_.
9. The term API stands for \_\_\_?

## Exercises

1. Create new classes for each real-world object that you observed at the beginning of this trail. Refer to the Bicycle class if you forget the required syntax.
2. For each new class that you've created above, create an interface that defines its behavior, then require your class to implement it. Omit one or two methods and try compiling. What does the error look like?

**Operators**

**Questions**

1. Consider the following code snippet.
2. arrayOfInts[j] > arrayOfInts[j+1]

Which operators does the code contain?

1. Consider the following code snippet.
2. int i = 10;
3. int n = i++%5;
   1. What are the values of i and n after the code is executed?
   2. What are the final values of i and n if instead of using the postfix increment operator (i++), you use the prefix version (++i))?
4. To invert the value of a boolean, which operator would you use?
5. Which operator is used to compare two values, = or == ?
6. Explain the following code sample: result = someCondition ? value1 : value2;

**Exercises**

1. Change the following program to use compound assignments:
2. class ArithmeticDemo {
3. public static void main (String[] args){
5. int result = 1 + 2; // result is now 3
6. System.out.println(result);
7. result = result - 1; // result is now 2
8. System.out.println(result);
9. result = result \* 2; // result is now 4
10. System.out.println(result);
11. result = result / 2; // result is now 2
12. System.out.println(result);
13. result = result + 8; // result is now 10
14. result = result % 7; // result is now 3
15. System.out.println(result);
16. }
17. }
18. In the following program, explain why the value "6" is printed twice in a row:
19. class PrePostDemo {
20. public static void main(String[] args){
21. int i = 3;
22. i++;
23. System.out.println(i); // "4"
24. ++i;
25. System.out.println(i); // "5"
26. System.out.println(++i); // "6"
27. System.out.println(i++); // "6"
28. System.out.println(i); // "7"
29. }

}

**Control Flow**

**Questions**

1. The most basic control flow statement supported by the Java programming language is the \_\_\_ statement.
2. The \_\_\_ statement allows for any number of possible execution paths.
3. The \_\_\_ statement is similar to the while statement, but evaluates its expression at the \_\_\_ of the loop.
4. How do you write an infinite loop using the for statement?
5. How do you write an infinite loop using the while statement?

**Exercises**

1. Consider the following code snippet.
2. if (aNumber >= 0)
3. if (aNumber == 0)
4. System.out.println("first string");
5. else System.out.println("second string");
6. System.out.println("third string");
   1. What output do you think the code will produce if aNumber is 3?
   2. Write a test program containing the previous code snippet; make aNumber 3. What is the output of the program? Is it what you predicted? Explain why the output is what it is; in other words, what is the control flow for the code snippet?
   3. Using only spaces and line breaks, reformat the code snippet to make the control flow easier to understand.
   4. Use braces, { and }, to further clarify the code.

**Expression**

# Expressions, Statements, and Blocks

## Questions

1. Operators may be used in building \_\_\_, which compute values.
2. Expressions are the core components of \_\_\_.
3. Statements may be grouped into \_\_\_.
4. The following code snippet is an example of a \_\_\_ expression.
5. 1 \* 2 \* 3
6. Statements are roughly equivalent to sentences in natural languages, but instead of ending with a period, a statement ends with a \_\_\_.
7. A block is a group of zero or more statements between balanced \_\_\_ and can be used anywhere a single statement is allowed.

## Exercises

Identify the following kinds of expression statements:

* aValue = 8933.234;
* aValue++;
* System.out.println("Hello World!");
* Bicycle myBike = new Bicycle();

**Interface**

1. Write a class that implements the CharSequence interface found in the java.lang package. Your implementation should return the string backwards. Select one of the sentences from this book to use as the data. Write a small main method to test your class; make sure to call all four methods.
2. Suppose you have written a time server that periodically notifies its clients of the current date and time. Write an interface the server could use to enforce a particular protocol on its clients.

**Interface Question**

1. What methods would a class that implements the java.lang.CharSequence interface have to implement?
2. What is wrong with the following interface?
3. public interface SomethingIsWrong {
4. void aMethod(int aValue){
5. System.out.println("Hi Mom");
6. }
7. }
8. Fix the interface in question 2.
9. Is the following interface valid?
10. public interface Marker {

}

**Inheritance**

**Questions**

1. Consider the following two classes:

public class ClassA {

public void methodOne(int i) {

}

public void methodTwo(int i) {

}

public static void methodThree(int i) {

}

public static void methodFour(int i) {

}

}

public class ClassB extends ClassA {

public static void methodOne(int i) {

}

public void methodTwo(int i) {

}

public void methodThree(int i) {

}

public static void methodFour(int i) {

}

}

a. Which method overrides a method in the superclass?  
b. Which method hides a method in the superclass?  
c. What do the other methods do?  
  
2. Consider the [Card](https://docs.oracle.com/javase/tutorial/java/IandI/examples/Card.java), [Deck](https://docs.oracle.com/javase/tutorial/java/IandI/examples/Deck.java), and [DisplayDeck](https://docs.oracle.com/javase/tutorial/java/IandI/examples/DisplayDeck.java" \t "_blank) classes you wrote in [Questions and Exercises: Classes](https://docs.oracle.com/javase/tutorial/java/javaOO/QandE/creating-questions.html). What Object methods should each of these classes override?

**Exercises**

* 1. Write the implementations for the methods that you answered in question 2.

# Control Flow Statements

**Questions**

1. The most basic control flow statement supported by the Java programming language is the \_\_\_ statement.
2. The \_\_\_ statement allows for any number of possible execution paths.
3. The \_\_\_ statement is similar to the while statement, but evaluates its expression at the \_\_\_ of the loop.
4. How do you write an infinite loop using the for statement?
5. How do you write an infinite loop using the while statement?

**Exercises**

1. Consider the following code snippet.
2. if (aNumber >= 0)
3. if (aNumber == 0)
4. System.out.println("first string");
5. else System.out.println("second string");
6. System.out.println("third string");
   1. What output do you think the code will produce if aNumber is 3?
   2. Write a test program containing the previous code snippet; make aNumber 3. What is the output of the program? Is it what you predicted? Explain why the output is what it is; in other words, what is the control flow for the code snippet?
   3. Using only spaces and line breaks, reformat the code snippet to make the control flow easier to understand.
   4. Use braces, { and }, to further clarify the code.

# Questions and Exercises: Enum Types

## Questions

1. True or false: an Enum type can be a subclass of java.lang.String.

## Exercises

1. Rewrite the class Card from the exercise in [Questions and Exercises: Classes](https://docs.oracle.com/javase/tutorial/java/javaOO/QandE/creating-questions.html) so that it represents the rank and suit of a card with enum types.
2. Rewrite the Deck class.

# Questions and Exercises: Annotations

## Questions

1. What is wrong with the following interface?
2. public interface House {
3. @Deprecated
4. void open();
5. void openFrontDoor();
6. void openBackDoor();
7. }
8. Consider this implementation of the House interface, shown in Question 1.
9. public class MyHouse implements House {
10. public void open() {}
11. public void openFrontDoor() {}
12. public void openBackDoor() {}
13. }

If you compile this program, the compiler produces a warning because open was deprecated (in the interface). What can you do to get rid of that warning?

1. Will the following code compile without error? Why or why not?
2. public @interface Meal { ... }
3. @Meal("breakfast", mainDish="cereal")
4. @Meal("lunch", mainDish="pizza")
5. @Meal("dinner", mainDish="salad")
6. public void evaluateDiet() { ... }

## Exercises

1. Define an annotation type for an enhancement request with elements id, synopsis, engineer, and date. Specify the default value as unassigned for engineer and unknown for date.

# Questions and Exercises: Inheritance

## Questions

1. Consider the following two classes:

public class ClassA {

public void methodOne(int i) {

}

public void methodTwo(int i) {

}

public static void methodThree(int i) {

}

public static void methodFour(int i) {

}

}

public class ClassB extends ClassA {

public static void methodOne(int i) {

}

public void methodTwo(int i) {

}

public void methodThree(int i) {

}

public static void methodFour(int i) {

}

}

a. Which method overrides a method in the superclass?  
b. Which method hides a method in the superclass?  
c. What do the other methods do?  
  
2. Consider the [Card](https://docs.oracle.com/javase/tutorial/java/IandI/examples/Card.java), [Deck](https://docs.oracle.com/javase/tutorial/java/IandI/examples/Deck.java), and [DisplayDeck](https://docs.oracle.com/javase/tutorial/java/IandI/examples/DisplayDeck.java" \t "_blank) classes you wrote in [Questions and Exercises: Classes](https://docs.oracle.com/javase/tutorial/java/javaOO/QandE/creating-questions.html). What Object methods should each of these classes override?

## Exercises

1. Write the implementations for the methods that you answered in question 2.

# Questions and Exercises: Characters and Strings

## Questions

1. What is the initial capacity of the following string builder?
2. StringBuilder sb = new StringBuilder("Able was I ere I saw Elba.");
3. Consider the following string:
4. String hannah = "Did Hannah see bees? Hannah did.";
   1. What is the value displayed by the expression hannah.length()?
   2. What is the value returned by the method call hannah.charAt(12)?
   3. Write an expression that refers to the letter b in the string referred to by hannah.
5. How long is the string returned by the following expression? What is the string?
6. "Was it a car or a cat I saw?".substring(9, 12)
7. In the following program, called [ComputeResult](https://docs.oracle.com/javase/tutorial/java/data/QandE/ComputeResult.java" \t "_blank), what is the value of result after each numbered line executes?
8. public class ComputeResult {
9. public static void main(String[] args) {
10. String original = "software";
11. StringBuilder result = new StringBuilder("hi");
12. int index = original.indexOf('a');
13. /\*1\*/ result.setCharAt(0, original.charAt(0));
14. /\*2\*/ result.setCharAt(1, original.charAt(original.length()-1));
15. /\*3\*/ result.insert(1, original.charAt(4));
16. /\*4\*/ result.append(original.substring(1,4));
17. /\*5\*/ result.insert(3, (original.substring(index, index+2) + " "));
18. System.out.println(result);
19. }
20. }

## Exercises

1. Show two ways to concatenate the following two strings together to get the string "Hi, mom.":
2. String hi = "Hi, ";
3. String mom = "mom.";
4. Write a program that computes your initials from your full name and displays them.
5. An anagram is a word or a phrase made by transposing the letters of another word or phrase; for example, "parliament" is an anagram of "partial men," and "software" is an anagram of "swear oft." Write a program that figures out whether one string is an anagram of another string. The program should ignore white space and punctuation.

# Questions and Exercises: Generics

1. Write a generic method to count the number of elements in a collection that have a specific property (for example, odd integers, prime numbers, palindromes).
2. Will the following class compile? If not, why?
3. public final class Algorithm {
4. public static <T> T max(T x, T y) {
5. return x > y ? x : y;
6. }
7. }
8. Write a generic method to exchange the positions of two different elements in an array.
9. If the compiler erases all type parameters at compile time, why should you use generics?
10. What is the following class converted to after type erasure?
11. public class Pair<K, V> {
12. public Pair(K key, V value) {
13. this.key = key;
14. this.value = value;
15. }
16. public K getKey(); { return key; }
17. public V getValue(); { return value; }
18. public void setKey(K key) { this.key = key; }
19. public void setValue(V value) { this.value = value; }
20. private K key;
21. private V value;
22. }
23. What is the following method converted to after type erasure?
24. public static <T extends Comparable<T>>
25. int findFirstGreaterThan(T[] at, T elem) {
26. // ...
27. }
28. Will the following method compile? If not, why?
29. public static void print(List<? extends Number> list) {
30. for (Number n : list)
31. System.out.print(n + " ");
32. System.out.println();
33. }
34. Write a generic method to find the maximal element in the range [begin, end) of a list.
35. Will the following class compile? If not, why?
36. public class Singleton<T> {
37. public static T getInstance() {
38. if (instance == null)
39. instance = new Singleton<T>();
40. return instance;
41. }
42. private static T instance = null;
43. }
44. Given the following classes:
45. class Shape { /\* ... \*/ }
46. class Circle extends Shape { /\* ... \*/ }
47. class Rectangle extends Shape { /\* ... \*/ }
48. class Node<T> { /\* ... \*/ }

Will the following code compile? If not, why?

Node<Circle> nc = new Node<>();

Node<Shape> ns = nc;

1. Consider this class:
2. class Node<T> implements Comparable<T> {
3. public int compareTo(T obj) { /\* ... \*/ }
4. // ...
5. }

Will the following code compile? If not, why?

Node<String> node = new Node<>();

Comparable<String> comp = node;

1. How do you invoke the following method to find the first integer in a list that is relatively prime to a list of specified integers?
2. public static <T>
3. int findFirst(List<T> list, int begin, int end, UnaryPredicate<T> p)

Note that two integers a and b are relatively prime if gcd(a, b) = 1, where gcd is short for greatest common divisor.

# Questions and Exercises: Creating and Using Packages

## Questions

Assume you have written some classes. Belatedly, you decide they should be split into three packages, as listed in the following table. Furthermore, assume the classes are currently in the default package (they have no package statements).

|  |  |
| --- | --- |
| **Destination Packages** | |
| **Package Name** | **Class Name** |
| mygame.server | Server |
| mygame.shared | Utilities |
| mygame.client | Client |

1. Which line of code will you need to add to each source file to put each class in the right package?
2. To adhere to the directory structure, you will need to create some subdirectories in the development directory and put source files in the correct subdirectories. What subdirectories must you create? Which subdirectory does each source file go in?
3. Do you think you'll need to make any other changes to the source files to make them compile correctly? If so, what?

## Exercises

Download the source files as listed here.

* [Client](https://docs.oracle.com/javase/tutorial/java/package/QandE/question/Client.java)
* [Server](https://docs.oracle.com/javase/tutorial/java/package/QandE/question/Server.java)
* [Utilities](https://docs.oracle.com/javase/tutorial/java/package/QandE/question/Utilities.java)

1. Implement the changes you proposed in questions 1 through 3 using the source files you just downloaded.
2. Compile the revised source files. (*Hint:* If you're invoking the compiler from the command line (as opposed to using a builder), invoke the compiler from the directory that contains the mygame directory you just created.)

# Questions and Exercises

## Questions

1. Is the following code legal?
2. try {
4. } finally {
6. }
7. What exception types can be caught by the following handler?
8. catch (Exception e) {
10. }

What is wrong with using this type of exception handler?

1. Is there anything wrong with the following exception handler as written? Will this code compile?
2. try {
3. } catch (Exception e) {
5. } catch (ArithmeticException a) {
7. }
8. Match each situation in the first list with an item in the second list.
   1. int[] A;  
      A[0] = 0;
   2. The JVM starts running your program, but the JVM can't find the Java platform classes. (The Java platform classes reside in classes.zip or rt.jar.)
   3. A program is reading a stream and reaches the end of stream marker.
   4. Before closing the stream and after reaching the end of stream marker, a program tries to read the stream again.
   5. \_\_error
   6. \_\_checked exception
   7. \_\_compile error
   8. \_\_no exception

## Exercises

* 1. Add a readList method to [ListOfNumbers.java](https://docs.oracle.com/javase/tutorial/essential/exceptions/examples/ListOfNumbers.java). This method should read in int values from a file, print each value, and append them to the end of the vector. You should catch all appropriate errors. You will also need a text file containing numbers to read in.
  2. Modify the following cat method so that it will compile.
  3. public static void cat(File file) {
  4. RandomAccessFile input = null;
  5. String line = null;
  6. try {
  7. input = new RandomAccessFile(file, "r");
  8. while ((line = input.readLine()) != null) {
  9. System.out.println(line);
  10. }
  11. return;
  12. } finally {
  13. if (input != null) {
  14. input.close();
  15. }
  16. }

}

# Questions and Exercises: Basic I/O

## Questions

1. What class and method would you use to read a few pieces of data that are at known positions near the end of a large file?

2. When invoking format, what is the best way to indicate a new line?

3. How would you determine the MIME type of a file?

4. What method(s) would you use to determine whether a file is a symbolic link?

## Exercises

1. Write an example that counts the number of times a particular character, such as e, appears in a file. The character can be specified at the command line. You can use [xanadu.txt](https://docs.oracle.com/javase/tutorial/essential/io/examples/xanadu.txt) as the input file.

2. The file [datafile](https://docs.oracle.com/javase/tutorial/essential/io/QandE/datafile" \t "_blank) begins with a single long that tells you the offset of a single int piece of data within the same file. Write a program that gets the int piece of data. What is the int data?

# Questions and Exercises: Concurrency

## Questions

1. Can you pass a Thread object to Executor.execute? Would such an invocation make sense?

## Exercises

1. Compile and run [BadThreads.java](https://docs.oracle.com/javase/tutorial/essential/concurrency/QandE/BadThreads.java):
2. public class BadThreads {
3. static String message;
4. private static class CorrectorThread
5. extends Thread {
6. public void run() {
7. try {
8. sleep(1000);
9. } catch (InterruptedException e) {}
10. // Key statement 1:
11. message = "Mares do eat oats.";
12. }
13. }
14. public static void main(String args[])
15. throws InterruptedException {
16. (new CorrectorThread()).start();
17. message = "Mares do not eat oats.";
18. Thread.sleep(2000);
19. // Key statement 2:
20. System.out.println(message);
21. }
22. }

The application should print out "Mares do eat oats." Is it guaranteed to always do this? If not, why not? Would it help to change the parameters of the two invocations of Sleep? How would you guarantee that all changes to message will be visible in the main thread?

1. Modify the producer-consumer example in [Guarded Blocks](https://docs.oracle.com/javase/tutorial/essential/concurrency/guardmeth.html) to use a standard library class instead of the Drop class.

# Questions and Exercises: Regular Expressions

## Questions

1. What are the three public classes in the java.util.regex package? Describe the purpose of each.
2. Consider the string literal "foo". What is the start index? What is the end index? Explain what these numbers mean.
3. What is the difference between an ordinary character and a metacharacter? Give an example of each.
4. How do you force a metacharacter to act like an ordinary character?
5. What do you call a set of characters enclosed in square brackets? What is it for?
6. Here are three predefined character classes: \d, \s, and \w. Describe each one, and rewrite it using square brackets.
7. For each of \d, \s, and \w, write *two* simple expressions that match the *opposite* set of characters.
8. Consider the regular expression (dog){3}. Identify the two subexpressions. What string does the expression match?

## Exercises

1. Use a backreference to write an expression that will match a person's name only if that person's first name and last name are the same.

# Questions and Exercises: Aggregate Operations

## Questions

1. A sequence of aggregate operations is known as a \_\_\_ .
2. Each pipeline contains zero or more \_\_\_ operations.
3. Each pipeline ends with a \_\_\_ operation.
4. What kind of operation produces another stream as its output?
5. Describe one way in which the forEach aggregate operation differs from the enhanced for statement or iterators.
6. True or False: A stream is similar to a collection in that it is a data structure that stores elements.
7. Identify the intermediate and terminal operations in this code:
8. double average = roster
9. .stream()
10. .filter(p -> p.getGender() == Person.Sex.MALE)
11. .mapToInt(Person::getAge)
12. .average()
13. .getAsDouble();
14. The code p -> p.getGender() == Person.Sex.MALE is an example of what?
15. The code Person::getAge is an example of what?
16. Terminal operations that combine the contents of a stream and return one value are known as what?
17. Name one important difference between the Stream.reduce method and the Stream.collect method.
18. If you wanted to process a stream of names, extract the male names, and store them in a new List, would Stream.reduce or Stream.collect be the most appropriate operation to use?
19. True or False: Aggregate operations make it possible to implement parallelism with non-thread-safe collections.
20. Streams are always serial unless otherwise specified. How do you request that a stream be processed in parallel?

## Exercises

1. Write the following enhanced for statement as a pipeline with lambda expressions. Hint: Use the filter intermediate operation and the forEach terminal operation.
2. for (Person p : roster) {
3. if (p.getGender() == Person.Sex.MALE) {
4. System.out.println(p.getName());
5. }
6. }
7. Convert the following code into a new implementation that uses lambda expressions and aggregate operations instead of nested for loops. Hint: Make a pipeline that invokes the filter, sorted, and collect operations, in that order.
8. List<Album> favs = new ArrayList<>();
9. for (Album a : albums) {
10. boolean hasFavorite = false;
11. for (Track t : a.tracks) {
12. if (t.rating >= 4) {
13. hasFavorite = true;
14. break;
15. }
16. }
17. if (hasFavorite)
18. favs.add(a);
19. }
20. Collections.sort(favs, new Comparator<Album>() {
21. public int compare(Album a1, Album a2) {
22. return a1.name.compareTo(a2.name);

}});

# Questions and Exercises: Implementations

## Questions

1. You plan to write a program that uses several basic collection interfaces: Set, List, Queue, and Map. You're not sure which implementations will work best, so you decide to use general-purpose implementations until you get a better idea how your program will work in the real world. Which implementations are these?
2. If you need a Set implementation that provides value-ordered iteration, which class should you use?
3. Which class do you use to access wrapper implementations?

## Exercises

1. Write a program that reads a text file, specified by the first command line argument, into a List. The program should then print random lines from the file, the number of lines printed to be specified by the second command line argument. Write the program so that a correctly-sized collection is allocated all at once, instead of being gradually expanded as the file is read in. Hint: To determine the number of lines in the file, use [java.io.File.length](https://docs.oracle.com/javase/8/docs/api/java/io/File.html" \l "length--" \t "_blank) to obtain the size of the file, then divide by an assumed size of an average line.

# Questions and Exercises: Interfaces

## Questions

1. Two distinct inheritance trees. One interface in particular is not considered to be a true Collection, and therefore sits at the top of its own tree. What is the name of this interface?
2. Each interface in the collections framework is declared with the <E> syntax, which tells you that it is generic. When you declare a Collection instance, what is the advantage of specifying the type of objects that it will contain?
3. What interface represents a collection that does not allow duplicate elements?
4. What interface forms the root of the collections hierarchy?
5. What interface represents an ordered collection that may contain duplicate elements?
6. What interface represents a collection that holds elements prior to processing?
7. What interface repesents a type that maps keys to values?
8. Name three different ways to iterate over the elements of a List.
9. True or False: Aggregate operations are mutative operations that modify the underlying collection.

## Exercises

1. Write a program that prints its arguments in random order. Do not make a copy of the argument array. Demonstrate how to print out the elements using both streams and the traditional enhanced for statement.
2. Take the [FindDups](https://docs.oracle.com/javase/tutorial/collections/interfaces/examples/FindDups.java" \t "_blank)example and modify it to use a SortedSet instead of a Set. Specify a Comparator so that case is ignored when sorting and identifying set elements.
3. Write a method that takes a List<String> and applies [String.trim](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html" \l "trim--" \t "_blank) to each element.
4. Consider the four core interfaces, Set, List, Queue, and Map. For each of the following four assignments, specify which of the four core interfaces is best-suited, and explain how to use it to implement the assignment.
   1. Whimsical Toys Inc (WTI) needs to record the names of all its employees. Every month, an employee will be chosen at random from these records to receive a free toy.
   2. WTI has decided that each new product will be named after an employee but only first names will be used, and each name will be used only once. Prepare a list of unique first names.
   3. WTI decides that it only wants to use the most popular names for its toys. Count up the number of employees who have each first name.
   4. WTI acquires season tickets for the local lacrosse team, to be shared by employees. Create a waiting list for this popular sport.