



OCA / OCP Java SE 8 Programmer Practice Tests



PREV

Chapter 16 Exceptions and Assertions



NEXT



Chapter 18 Java I/O Fundamentals

Chapter 17

Use Java SE 8 Date/Time API

THE OCP EXAM TOPICS COVERED IN THIS PRACTICE TEST INCLUDE THE FOLLOWING:

- ☒ Use Java SE 8 Date/Time API
 - Create and manage date-based and time-based events including a combination of date and time into a single object using `LocalDate`, `LocalTime`, `LocalDateTime`, `Instant`, `Period`, and `Duration`
 - Work with dates and times across timezones and manage changes resulting from daylight savings including Format date and times values
 - Define and create and manage date-based and time-based events using `Instant`, `Period`, `Duration`, and `TemporalUnit`
1. What package is the `LocalTime` class in?
 1. `java.date`
 2. `java.lang`
 3. `java.time`
 4. `java.util`
 2. How many of the classes `Duration`, `LocalDateTime`, and `LocalTime` have the concept of a time zone?
 1. None
 2. One
 3. Two
 4. Three
 3. Which class has a `getSeconds()` method?
 1. Only the `Duration` class
 2. Only the `Period` class
 3. Both the `Duration` and `Period` classes
 4. Neither class
 4. Which of these represents the earliest date/time?
 1. 2017-02-15T03:00+01:00[Europe/Berlin]
 2. 2017-02-15T04:00+02:00[Europe/Helsinki]
 3. 2017-02-15T05:00+01:00[Europe/Warsaw]
 4. None of the above. We have a tie.

You have 2 days left in your trial, Gtucker716. Subscribe today. [See pricing options.](#)

following code output?

```

LocalDate localDate = LocalDate.of(2017, 3, 12);
LocalTime localTime = LocalTime.of(1, 0);
ZoneId zone = ZoneId.of("America/New_York");
ZonedDateTime z = ZonedDateTime.of(localDate, localTime, zone);
Duration duration = Duration.ofHours(3);
ZonedDateTime later = z.plus(duration);
System.out.println(later.getHour());

```

1. 4
2. 5
3. 6
4. None of the above

6. What does the following output?

```

int year = 1874;
int month = Month.MARCH;
int day = 24;
LocalDate date = LocalDate.of(year, month, day);
System.out.println(date.isBefore(LocalDate.now()));

```

1. false
2. true
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

7. Which correctly fills in the blank to print 2017-01-15?

```

LocalDate hatDay = LocalDate.of(2017, Month.JANUARY, 15);
DateTimeFormatter f = DateTimeFormatter.ISO_DATE;
System.out.println(_____);

```

1. f.format(hatDay)
2. f.formatDate(hatDay)
3. hatDay.format(f)

1. I
2. III
3. I and III
4. II and III

8. Which of the answer choices is true given the following?

```

2017-01-07T10:00-07:00[America/Phoenix]
2017-01-07T08:00-08:00[America/Vancouver]

```

1. The first date/time is one hour earlier than the second.
 2. The first date/time is three hours earlier than the second.
 3. The first date/time is one hour later than the second.
 4. The first date/time is three hours later than the second.
9. Given that daylight savings time starts on March 12, 2017, at 2 a.m. and clocks jump from 1:59 a.m. to 03:00 a.m., which of the following can fill in the blank so the code doesn't throw an exception?

```

LocalDate localDate = LocalDate.of(2017, 3, 12);
LocalTime localTime = LocalTime.of(____);
ZoneId zone = ZoneId.of("America/New_York");
ZonedDateTime z = ZonedDateTime.of(localDate, localTime, zone);

```

1. 2, 0
2. 3, 0
3. Either of the above will run without throwing an exception.
4. Both of these will cause an exception to be thrown.

10. What is the result of the following?

```

11: LocalDate waffleDay = LocalDate.of(2017, Month.MARCH, 25);
12: Period period = Period.of(1, 6, 3);
13: LocalDate later = waffleDay.plus(period);
14: later.plusDays(1);
15: LocalDate thisOne = LocalDate.of(2018, Month.SEPTEMBER, 28);
16: LocalDate thatOne = LocalDate.of(2018, Month.SEPTEMBER, 29);
17: System.out.println(later.isBefore(thisOne) + " "
18:     + later.isBefore(thatOne));

```

1. false false
2. false true
3. true true
4. The code does not compile.

11. What is a possible result of the following?

```

LocalDate montyPythonDay = LocalDate.of(2017, Month.MAY, 10);
LocalDate aprilFools = LocalDate.of(2018, Month.APRIL, 1);
Duration duration = Duration.ofDays(1);
LocalDate result = montyPythonDay.minus(duration);
System.out.println(result + " " + aprilFools.isBefore(result));

```

1. 2017-05-09 false
2. 2017-05-09 true
3. The code does not compile.
4. None of the above

12. What is the result of running this code?

```

12: LocalDate pieDay = LocalDate.of(2017, Month.JANUARY, 23);
13: LocalTime midnight = LocalTime.of(0, 0);
14: LocalDateTime pieTime = LocalDateTime.of(pieDay, midnight);
15:
16: DateTimeFormatter f = DateTimeFormatter
17:     .ofLocalizedDate(FormatStyle.SHORT);
18: f.format(pieDay);
19: f.format(pieTime);
20: f.format(midnight);

```

1. The code runs successfully.
2. The code throws an exception on line 19.
3. The code throws an exception on line 20.
4. The code does not compile.

13. In the United States, daylight savings time ends on November 5th, 2017 at 02:00 a.m. and we repeat the previous hour. What is the output of the following?

```

import java.time.*;

public class FallBack {

    public static void main(String[] args) {
        LocalDate localDate = LocalDate.of(2017, Month.NOVEMBER, 5);
        LocalTime localTime = LocalTime.of(1, 0);
        ZoneId zone = ZoneId.of("America/New_York");
        ZonedDateTime z = ZonedDateTime.of(localDate, localTime, zone

        for (int i = 0; i < 6; i++)
            z.plusHours(1);

        System.out.println(z.getHour());
    }
}

```

1. 5
2. 6
3. 7
4. None of the above

14. What format pattern would you pass to a `DateTimeFormatter` so it creates hour and minute output such as 02:33?

1. HH:MM
2. HH:mm
3. hh:MM
4. hh:mm

15. `LocalTime.of()` has a number of overloads. Which of the following is not one of them?

1. `LocalTime.of(int hour, int minute)`
2. `LocalTime.of(int hour, int minute, int second)`
3. `LocalTime.of(int hour, int minute, int second, int nanoOfSecond)`
4. `LocalTime.of(int hour, int minute, int second, int nanoOfSecond, int picoSeconds)`

16. How many of the classes `LocalDate`, `Period`, and `ZonedDateTime` have a method to get the year?

1. None
2. One
3. Two
4. Three

17. Which statement is not true about these two variables?

```
Duration duration = Duration.ofDays(1);
Period period = Period.ofDays(1);
```

1. Both output the same value when calling `toString()`.
2. The `Duration` object compiles because durations are for smaller units of time.
3. The `Period` object compiles because periods are for larger units of time.
4. None of the above

18. What is a possible output of this code?

```
LocalTime time = LocalTime.of(1,2,3,4);
System.out.println(time);
```

1. 01:02:03.4
2. 01:02:03.000000004
3. 01/01/1970 01:02:03.4
4. 01/01/1970 01:02:03.000000004

19. What does the following print?

```
import java.time.*;
import java.time.format.*;

public class PolarBear {
    public static void main(String[] args) {
        LocalDate polarBearDay = LocalDate.of(2017, 2, 27);
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yy:
        System.out.println(polarBearDay.format(formatter));
    }
}
```

1. 2017 27 Jan
2. 2017 27 Feb
3. 2017 Jan 27
4. 2017 Feb 27

20. Which contains a constant named `HOURS`?

1. `ChronoUnit`
2. `Duration`
3. `Instant`
4. `Period`

21. The United States observes daylight savings time on March 12, 2017, by moving the clocks forward an hour at 2 a.m. What does the following code output?

```
LocalDate localDate = LocalDate.of(2017, 3, 12);
LocalTime localTime = LocalTime.of(13, 0);
ZoneId zone = ZoneId.of("America/New_York");
ZonedDateTime z = ZonedDateTime.of(localDate, localTime, zone);
Duration duration = Duration.ofHours(3);
ZonedDateTime later = z.plus(duration);
System.out.println(later.getHour());
```

1. 13
2. 16
3. 17
4. None of the above

22. What is a possible result of the following?

```
LocalDate montyPythonDay = LocalDate.of(2017, Month.MAY, 10);
LocalTime time = LocalTime.of(5, 40);
LocalDateTime dateTime = LocalDateTime.of(montyPythonDay, time);
Duration duration = Duration.ofDays(1);
LocalDateTime result = dateTime.minus(duration);
System.out.println(result);
```

1. 2017-05-09
2. 2017-05-09T05:40
3. 2017-05-10T05:40
4. None of the above

23. Which correctly fills in the blank to print 2017-01-15?

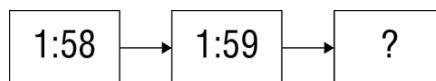
```
LocalDate hatDay = LocalDate.of(2017, Month.JANUARY, 15);
DateFormatter f = DateFormatter.ISO_DATE;
System.out.println(_____);
```

1. `f.format(hatDate)`
2. `hatDay.format(f)`
3. Both of the above
4. Neither of the above

24. `LocalDateTime.of()` has a number of overloads. Which of the following is not one of them?

1. `LocalDateTime.of(LocalDate date, LocalTime time)`
2. `LocalDateTime.of(LocalDate date, int hour, int minute)`
3. `LocalDateTime.of(int year, int month, int day, int hour, int minute)`
4. `LocalDateTime.of(int year, Month month, int day, int hour, int minute)`

25. In the United States, daylight savings time for 2017 starts at 2 a.m. on March 12th and ends at 2 a.m. on November 5th. Given the sequence in the following image, what time comes next on March 12th, July 4th, and November 5th, respectively?



1. 01:00, 02:00, 01:00

2. 01:00, 02:00, 03:00
3. 03:00, 02:00, 01:00
4. 03:00, 02:00, 03:00

26. What is the output of the following?

```
LocalDate date1 = LocalDate.of(2017, Month.MARCH, 3);
LocalDate date2 = LocalDate.of(2017, Month.FEBRUARY, 31);
System.out.println(date1.equals(date2));
```

1. false
2. true
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

27. Given this date/time and time zone offset, what time is it in GMT?

```
2017-03-09T16:00-10:00[US/Hawaii]
```

1. 02:00
2. 04:00
3. 06:00
4. 10:00

28. What is a possible output of the following?

```
LocalDate trainDay = LocalDate.of(2017, 5, 13);
LocalTime time = LocalTime.of(10, 0);
ZoneId zone = ZoneId.of("America/Los_Angeles");
ZonedDateTime zdt = ZonedDateTime.of(trainDay, time, zone);
Instant instant = zdt.toInstant();
instant = instant.plus(1, ChronoUnit.DAYS);
System.out.println(instant);
```

1. 2017-05-13T10:00-07:00[America/Los_Angeles]
2. 2017-05-13T17:00:00Z
3. 2017-05-14T10:00-07:00[America/Los_Angeles]
4. 2017-05-14T17:00:00Z

29. What is the output of the following?

```
LocalDate date = LocalDate.of(2017, Month.JULY, 17);
LocalTime time = LocalTime.of(10, 0);
ZoneId zone = ZoneId.of("America/New_York");
ZonedDateTime iceCreamDay = ZonedDateTime.of(date, time, zone);
time = time.plusMonths(1);
System.out.println(iceCreamDay.getMonthValue());
```

1. 6
2. 7
3. 8
4. The code does not compile.

30. What does the following print?

```
import java.time.*;
import java.time.format.*;

public class PolarBear {
    public static void main(String[] args) {
        LocalDate polarBearDay = LocalDate.of(2017, 2, 27);
        DateTimeFormatter formatter = DateTimeFormatter
            .ofPattern("Holiday: yyyy dd MMM");
        System.out.println(polarBearDay.format(formatter));
    }
}
```

1. Holiday: 2017 27 Jan
2. Holiday: 2017 27 Feb
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

31. Which of these represents the earliest date/time?

1. 2017-02-15T16:00+07:00[Asia/Bangkok]
2. 2017-02-15T18:00+04:00[Asia/Dubai]
3. 2017-02-15T20:00+08:00[Asia/Kuala_Lumpur]
4. None of the above. We have a tie.

32. What is the result of the following?

```
11: LocalDate waffleDay = LocalDate.of(2017, Month.MARCH, 25);
12: Period period = Period.ofYears(1).ofMonths(6).ofDays(3);
13: LocalDate later = waffleDay.plus(period);
14: later.plusDays(1); 15: LocalDate thisOne = LocalDate.of(2018,
16: LocalDate thatOne = LocalDate.of(2018, Month.SEPTEMBER, 29);
17: System.out.println(later.isBefore(thisOne) + " "
18:     + later.isBefore(thatOne));
```

1. false false
2. false true
3. true true
4. The code does not compile.

33. How many of the following can fill in the blank so this code compiles and prints 31?

```
LocalDate xmas = LocalDate.of(2017, 12, 25);
LocalDate blackFriday = LocalDate.of(2017, 11, 24);
long shoppingDaysLeft = _____;
System.out.println(shoppingDaysLeft);
```

1. blackFriday.until(xmas, ChronoUnit.DAYS)
 2. blackFriday.until(xmas, TemporalUnit.DAYS)
 3. ChronoUnit.DAYS.between(blackFriday, xmas)
 4. TemporalUnit.DAYS.between(blackFriday, xmas)
1. One
 2. Two
 3. Three
 4. Four

34. How many of these classes cause a compiler error when filling in the blank: LocalDate, LocalDateTime, LocalTime, ZonedDateTime?

```
private static String formatMe(_____ obj) {
    DateTimeFormatter f = DateTimeFormatter.ofLocalizedTime(FormatSt
    return f.format(obj);
}
```

1. None
2. One
3. Two
4. Three

35. What is the output of the following?

```
LocalDate date = LocalDate.of(2017, Month.JULY, 17);
LocalTime time = LocalTime.of(10, 0);
ZoneId zone = ZoneId.of("America/New_York");
ZonedDateTime iceCreamDay = ZonedDateTime.of(date, time, zone);
date = date.plusMonths(1);
System.out.println(iceCreamDay.getMonthValue());
```

1. 6
2. 7
3. 8
4. The code does not compile.

36. Which of the following can fill in the blank to make this code compile?

```
public boolean isMyBirthday(LocalDate dateTime) {
    _____
    return now.getMonth() == dateTime.getMonth()
        && now.getDayOfMonth() == dateTime.getDayOfMonth();
}
```

1. `LocalDate now = LocalDate.now();`
2. `LocalDate now = new LocalDate();`
3. `ZonedDateTime now = ZonedDateTime.now();`
4. `ZonedDateTime now = new ZonedDateTime();`

37. What is the output of the following?

```
LocalDate date1 = LocalDate.of(2017, Month.MARCH, 3);
LocalDate date2 = date1.plusDays(2).minusDays(1).minusDays(1);
System.out.println(date1.equals(date2));
```

1. `false`
2. `true`
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

38. What is a possible output of the following?

```
LocalDate date = LocalDate.of(2017, 5, 13);
LocalTime time = LocalTime.of(10, 0);
LocalDateTime trainDay = LocalDateTime.of(date, time);
Instant instant = trainDay.toInstant();
instant = instant.plus(1, ChronoUnit.DAYS);
System.out.println(instant);
```

1. `2017-05-14T10:00-07:00[America/Los_Angeles]`
2. `2017-05-14T17:00:00Z`
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

39. What is the result of the following?

```
public class PiDay {
    public static void main(String[] args) {
        LocalDateTime pi = LocalDateTime.of(2017, 3, 14, 1, 59);
        DateTimeFormatter formatter = DateTimeFormatter
            .ofPattern("M.ddhhmm");
        System.out.println(formatter.format(pi));
    }
}
```

1. `3.140159`
2. `59.140103`
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

40. Daylight savings time ends on November 5, 2017 at 2 a.m. when we repeat the hour. Suppose we have a `ZonedDateTime` that outputs `2017-11-05T01:00-04:00[America/New_York]` when calling `toString()`. What is a possible value of the `ZonedDateTime` obtained by adding an hour to this value?

1. `2017-11-05T01:00-04:00[America/New_York]`
2. `2017-11-05T02:00-04:00[America/New_York]`
3. `2017-11-05T01:00-05:00[America/New_York]`
4. `2017-11-05T02:00-05:00[America/New_York]`



◀ PREV
[Chapter 16 Exceptions and Assertions](#)

NEXT ▶
[Chapter 18 Java I/O Fundamentals](#)

[Recommended](#) / [Queue](#) / [History](#) / [Topics](#) / [Tutorials](#) / [Settings](#) / [Get the App](#) / [Sign Out](#)
© 2017 Safari. [Terms of Service](#) / [Privacy Policy](#)