Due to the print book page limit, we cannot inlcude all good CheckPoint questions in the physical book. The CheckPoint on this Website may contain extra questions not printed in the book. The questions in some sections may have been reordered as a result. Nevertheless, it is easy to find the CheckPoint questions in the book on this Website. Please send suggestions and errata to Dr. Liang at y.daniel.liang@gmail.com. Indicate the book, edition, and question number in your email. Thanks!

Chapter 6 Check Point Questions

Section 6.4

▼6.4.1

What are the benefits of using a method?

At least three benefits: (1) Reuse code; (2) Reduce complexity; (3) Easy to maintain.

Hide Answer Read Answer

▼6.4.2

How do you define a method? How do you invoke a method?

See the Sections 6.2 and 6.3 on how to define and invoke methods.

Hide Answer Read Answer

▼6.4.3

How do you simplify the max method in Listing 6.1 using the conditional operator?

```
return (num1 > num2) ? num1 : num2;
```

Hide Answer Read Answer

▼6.4.4

True or false? A call to a method with a void return type is always a statement itself, but a call to a value-returning method cannot be a statement by itself.

True: a call to a method with a void return type is always a statement itself.
False: a call to a value-returning method is always a component of an expression. Note that a value-returning method can also be invoked as a statement in Java. In this case, the caller simply ignores the return value.

Hide Answer Read Answer

▼6.4.5

What is the return type of a main method?

void

Hide Answer Read Answer

▼6.4.6

What would be wrong with not writing a return statement in a value-returning method? Can you have a return statement in a void method? Does the return statement in the following method cause syntax errors?

```
public static void xMethod(double x, double y) {
  System.out.println(x + y);
  return x + y;
}
```

A syntax error occurs if a return statement is not used to return a value in a value-returning method. You can have a return statement in a void method, which simply exits the method. But a return statement cannot return a value such as return x + y in a void method.

Hide Answer Read Answer

▼6.4.7

Define the terms parameter, argument, and method signature.

See Section 6.2.

Hide Answer Read Answer

▼6.4.8

Write method headers (not the bodies) for the following methods:
a. Return a sales commission, given the sales amount and the commission rate.
b. Display the calendar for a month, given the month and year.
c. Return a square root of a number.
d. Test whether a number is even, and returning true if it is.
e. Display a message a specified number of times.
f. Return the monthly payment, given the loan amount, number of years, and annual interest rate.
g. Return the corresponding uppercase letter, given a lowercase letter.

(a) public static double getCommission(double salesAmount, double commissionRate)
(b) public static void printCalendar(int month, int year)
(c) public static double sqrt(double value)
(d) public static boolean isEven(int value)
(e) public static void printMessage(String message, int times)
(f) public static double monthlyPayment(double loan, int numberOfYears, double annualInterestRate)
(g) public static char getUpperCase(char letter)

Hide Answer Read Answer

▼6.4.9

Identify and correct the errors in the following program:

```
 1  public class Test {
 2    public static method1(int n, m) {
 3      n += m;
 4      method2(3.4);
 5    }
 6
 7    public static int method2(int n) {
 8      if (n > 0) return 1;
 9      else if (n == 0) return 0;
10      else if (n < 0) return -1;
11    }
12  }
```

Line 2: method1 is not defined correctly. It does not have a return type or void.
Line 2: type int should be declared for parameter m.
Line 7: parameter type for n should be double to match method2(3.4).
Line 11: if (n < 0) should be removed in method, otherwise a compile error is reported.

Hide Answer Read Answer

▼6.4.10

Reformat the following program according to the programming style and documentation guidelines proposed in Section 1.9, Programming Style and Documentation. Use the end-of-line brace style.

```
public class Test {
  public static double method(double i, double j)
  {
    while (i < j) {
      j--;
    }

    return j;
  }
}
```

```
public class Test {
  public static double xMethod(double i, double j) {
    while (i < j) {
      j--;
    }

    return j;
  }
}
```

Hide Answer Read Answer

Section 6.5

▼6.5.1

How is an argument passed to a method? Can the argument have the same name as its parameter?

You pass actual parameters by passing the right type of value in the right order. The actual parameter can have the same name as its formal parameter.

Hide Answer Read Answer

▼6.5.2

Identify and correct the errors in the following program:

```
1   public class Test {
2     public static void main(String[] args) {
3       nPrintln(5, "Welcome to Java!");
4     }
5
6     public static void nPrintln(String message, int n) {
7       int n = 1;
```

```
 8       for (int i = 0; i < n; i++)
 9         System.out.println(message);
10     }
11   }
```

Two errors: 1. the arguments are not passed in the right order. 2. n is a parameter, but is redeclared in the nPrintln method in line 7.

Hide Answer Read Answer

▼6.5.3

What is pass-by-value? Show the result of the following programs.

```
(a)
public class Test {
  public static void main(String[] args) {
    int max = 0;
    max(1, 2, max);
    System.out.println(max);
  }

  public static void max(
      int value1, int value2, int max) {
    if (value1 > value2)
      max = value1;
    else
      max = value2;
  }
}

(b)
public class Test {
  public static void main(String[] args) {
    int i = 1;
    while (i <= 6) {
      method1(i, 2);
      i++;
    }
  }

  public static void method1(
      int i, int num) {
    for (int j = 1; j <= i; j++) {
      System.out.print(num + " ");
      num *= 2;
    }

    System.out.println();
  }
}

(c)
public class Test {
  public static void main(String[] args) {
    // Initialize times
    int times = 3;
```

```
    System.out.println("Before the call,"
      + " variable times is " + times);

    // Invoke nPrintln and display times
    nPrintln("Welcome to Java!", times);
    System.out.println("After the call,"
      + " variable times is " + times);
  }

  // Print the message n times
  public static void nPrintln(
      String message, int n) {
    while (n > 0) {
      System.out.println("n = " + n);
      System.out.println(message);
      n--;
    }
  }
}
```

```
(d)
public class Test {
  public static void main(String[] args) {
    int i = 0;
    while (i <= 4) {
      method1(i);
      i++;
    }

    System.out.println("i is " + i);
  }

  public static void method1(int i) {
    do {
      if (i % 3 != 0)
        System.out.print(i + " ");
      i--;
    }
    while (i >= 1);

    System.out.println();
  }
}
```

"Pass by value" is to pass a copy of the value to the method.
(a) The output of the program is 0, because the variable max is not changed by invoking the method max.

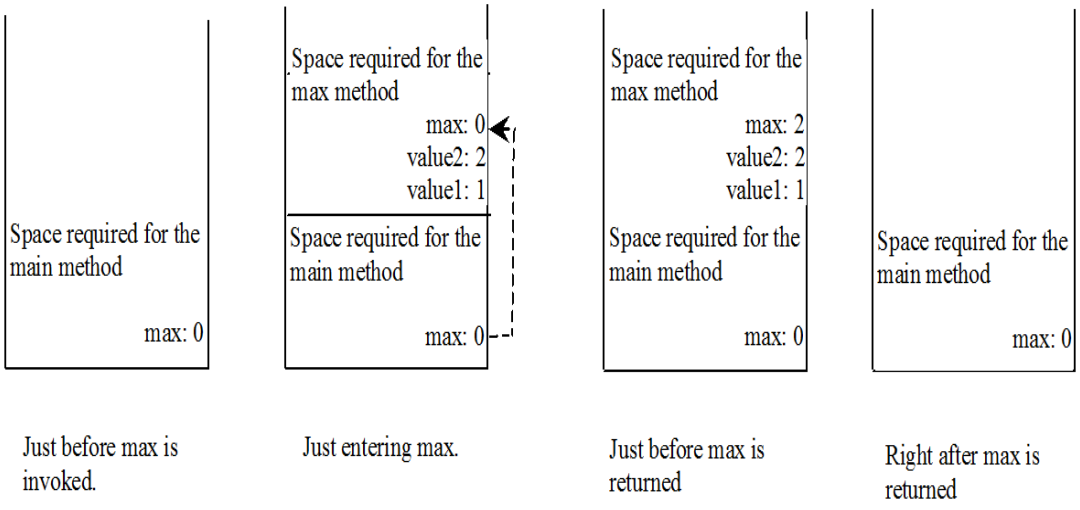
```
(b)
2
2 4
2 4 8
2 4 8 16
2 4 8 16 32
2 4 8 16 32 64
```

```
(c)
Before the call, variable times is 3
n = 3
Welcome to Java!
n = 2
Welcome to Java!
n = 1
Welcome to Java!
After the call, variable times is 3

(d)

1
2 1
2 1
4 2 1
i is 5
```

Hide Answer Read Answer

▼6.5.4

For (a) in the preceding question, show the contents of the activation records in the call stack just before the method max is invoked, just as max is entered, just before max is returned, and right after max is returned.



Hide Answer Read Answer

Section 6.6

▼6.6.1

Trace the gcd method to find the return value for gcd(4, 6).

gcd(4, 6) returns 2.

Hide Answer Read Answer

▼6.6.2

Trace the isPrime method to find the return value for isPrime(25).

isPrime(25) returns false.

Hide Answer Read Answer

Section 6.7

▼6.7.1

What is hexCharToDecimal('B')? What is hexCharToDecimal('7')? What is hexToDecimal('A9')?

hexCharToDecimal('B') returns 11. hexCharToDecimal('7') returns 7. hexToDecimal("A9") returns 169.

Hide Answer Read Answer

Section 6.8

▼6.8.1

What is method overloading? Is it permissible to define two methods that have the same name but different parameter types? Is it permissible to define two methods in a class that have identical method names and parameter lists but different return value types or different modifiers?

Two methods with the same name, defined in the same class, is called method overloading. It is fine to have same method name, but different parameter types. You cannot overload methods based on return type, or modifiers.

Hide Answer Read Answer

▼6.8.2

What is wrong in the following program?

```
public class Test {
  public static void method(int x) {
  }

  public static int method(int y) {
    return y;
  }
}
```

Methods public static void method(int x) and public static int method(int y) have the same signature method(int).

Hide Answer Read Answer

▼6.8.3

Given two method definitions,

```
public static double m(double x, double y)
public static double m(int x, double y)
```

tell which of the two methods is invoked for:

```
a.      double z = m(4, 5);
b.      double z = m(4, 5.4);
c.      double z = m(4.5, 5.4);
```

(a) invokes the second method. (b) invokes the second. (c) invokes the first method.

Hide Answer Read Answer

Section 6.9

▼6.9.1

What is a local variable?

A local variable is a variable declared inside a method.

Hide Answer Read Answer

▼6.9.2

What is the scope of a local variable?

The scope of a local variable starts from its declaration and continues to the end of the block that contains the variable.

Hide Answer Read Answer