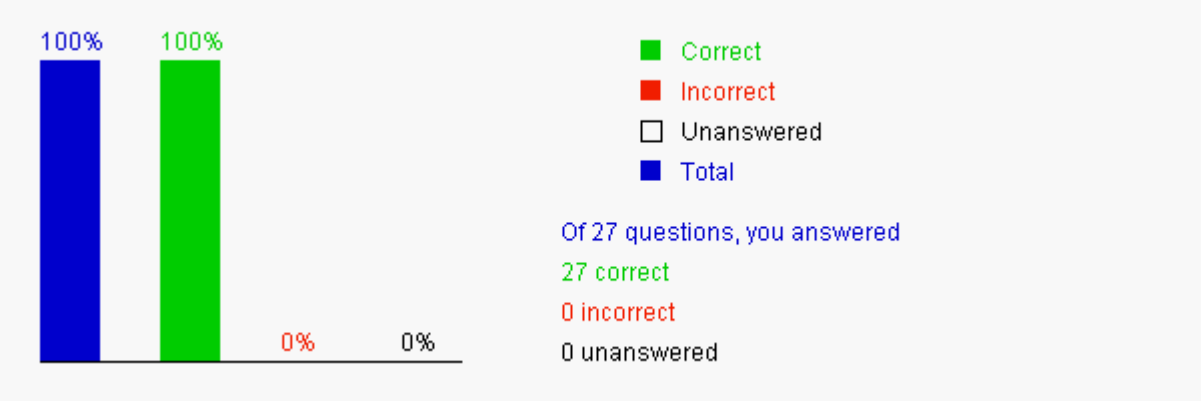
This quiz is for students to practice. A large number of additional quiz is available for instructors using Quiz Generator from the Instructor's Resource Website. Videos for Java, Python, and C++ can be found at https://yongdanielliang.github.io/revelvideos.html.

Chapter 6 Methods



Please send suggestions and errata to Dr. Liang at y.daniel.liang@gmail.com. Indicate which book and edition you are using. Thanks!

Sections 6.2 Defining a Method

6.1 Suppose your method does not return any value, which of the following keywords can be used as a return type?

- (•) A. void
- ( ) B. int
- ( ) C. double
- ( ) D. public
- ( ) E. None of the above

Your answer is correct

6.2 The signature of a method consists of _____.

- ( ) A. method name
- (•) B. method name and parameter list
- ( ) C. return type, method name, and parameter list
- ( ) D. parameter list

Your answer is correct
Explanation: By definition, a method signature consists of method name and parameter list.

6.3 All Java applications must have a method _____.

- ( ) A. public static Main(String[] args)
- ( ) B. public static Main(String args[])
- (•) C. public static void main(String[] args)
- ( ) D. public void main(String[] args)
- ( ) E. public static main(String[] args)

Your answer is correct
Explanation: Java application's starting method is the main method.

Sections 6.3 Calling a Method

6.4 Arguments to methods always appear within _____.

- ( ) A. brackets
- (•) B. parentheses
- ( ) C. curly braces
- ( ) D. quotation marks

Your answer is correct

6.5 Does the return statement in the following method cause compile errors?

```
public static void main(String[] args) {
  int max = 0;
  if (max != 0)
    System.out.println(max);
  else
```

```
    return;
}
```

- ○ A. Yes
- ◉ B. No

Your answer is correct 

Explanation: It is rare, but sometimes useful to have a return statement for circumventing the normal flow of control in a void method.

6.6 Does the method call in the following method cause compile errors?

```
public static void main(String[] args) {
  Math.pow(2, 4);
}
```

- ○ A. Yes
- ◉ B. No

Your answer is correct 

Explanation: A value-returning method can also be invoked as a statement in Java. In this case, the caller simply ignores the return value. This is rare, but permissible if the caller is not interested in the return value.

6.7 Each time a method is invoked, the system stores parameters and local variables in an area of memory, known as _____, which stores elements in last-in first-out fashion.

- ○ A. a heap
- ○ B. storage area
- ◉ C. a stack
- ○ D. an array

Your answer is correct 

Sections 6.4 void vs. Value-Returning Methods

6.8 Which of the following should be defined as a void method?

- ◉ A. Write a method that prints integers from 1 to 100.
- ○ B. Write a method that returns a random integer from 1 to 100.
- ○ C. Write a method that checks whether a number is from 1 to 100.
- ○ D. Write a method that converts an uppercase letter to lowercase.

Your answer is correct 

6.9 You should fill in the blank in the following code with _____.

```
public class Test {
  public static void main(String[] args) {
    System.out.print("The grade is ");
    printGrade(78.5);

    System.out.print("The grade is ");
    printGrade(59.5);
  }

  public static __________ printGrade(double score) {
    if (score >= 90.0) {
      System.out.println('A');
    }
    else if (score >= 80.0) {
      System.out.println('B');
    }
    else if (score >= 70.0) {
      System.out.println('C');
    }
    else if (score >= 60.0) {
      System.out.println('D');
    }
    else {
      System.out.println('F');
    }
  }
}
```

- ○ A. int
- ○ B. double
- ○ C. boolean
- ○ D. char
- ◉ E. void

Your answer is correct
Explanation: void should here because the method does not return any value.

6.10 You should fill in the blank in the following code with _____.

```
public class Test {
  public static void main(String[] args) {
    System.out.print("The grade is " + getGrade(78.5));
    System.out.print("\nThe grade is " + getGrade(59.5));
  }

  public static _________ getGrade(double score) {
    if (score >= 90.0)
      return 'A';
    else if (score >= 80.0)
      return 'B';
    else if (score >= 70.0)
      return 'C';
    else if (score >= 60.0)
      return 'D';
    else
      return 'F';
  }
}
```

- A. int
- B. double
- C. boolean
- D. char
- E. void

Your answer is correct
Explanation: char should be placed here because the method returns a character.

6.11 Consider the following incomplete code:

```
public class Test {
  public static void main(String[] args) {
    System.out.println(f(5));
  }

  public static int f(int number) {
    // Missing body
  }
}
```

The missing method body should be _____.

- A. return "number";
- B. System.out.println(number);
- C. System.out.println("number");
- D. return number;

Your answer is correct

Sections 6.5 Passing Parameters by Values

6.12 When you invoke a method with a parameter, the value of the argument is passed to the parameter. This is referred to as _____.

- A. method invocation
- B. pass by value
- C. pass by reference
- D. pass by name

Your answer is correct

6.13 Given the following method, what is the output of the call nPrint('a', 4)?

```
static void nPrint(String message, int n) {
  while (n > 0) {
    System.out.print(message);
    n--;
  }
}
```

- A. aaaaa

B. aaaa
C. aaa
D. invalid call

Your answer is correct
Explanation: Invalid call because char 'a' cannot be passed to string message

6.14 Given the following method

```
static void nPrint(String message, int n) {
  while (n > 0) {
    System.out.print(message);
    n--;
  }
}
```

What is k after invoking nPrint("A message", k)?

```
int k = 2;
nPrint("A message", k);
```

A. 0
B. 1
C. 2
D. 3

Your answer is correct

Section 6.8 Overloading Methods

6.15 Analyze the following code:

```
public class Test {
  public static void main(String[] args) {
    System.out.println(xMethod(5, 500L));
  }

  public static int xMethod(int n, long l) {
    System.out.println("int, long");
    return n;
  }

  public static long xMethod(long n, long l) {
    System.out.println("long, long");
    return n;
  }
}
```

A. The program displays int, long followed by 5.
B. The program displays long, long followed by 5.
C. The program runs fine but displays things other than 5.
D. The program does not compile because the compiler cannot distinguish which xmethod to invoke.

Your answer is correct

6.16 Analyze the following code:

```
class Test {
  public static void main(String[] args) {
    System.out.println(xmethod(5));
  }

  public static int xmethod(int n, long t) {
    System.out.println("int");
    return n;
  }

  public static long xmethod(long n) {
    System.out.println("long");
    return n;
  }
}
```

A. The program displays int followed by 5.
B. The program displays long followed by 5.
C. The program runs fine but displays things other than 5.

D. The program does not compile because the compiler cannot distinguish which xmethod to invoke.

Your answer is correct

6.17 Analyze the following code.

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

  public static double max(int num1, double num2) {
    System.out.println("max(int, double) is invoked");

    if (num1 > num2)
      return num1;
    else
      return num2;
  }

  public static double max(double num1, int num2) {
    System.out.println("max(double, int) is invoked");

    if (num1 > num2)
      return num1;
    else
      return num2;
  }
}
```

A. The program cannot compile because you cannot have the print statement in a non-void method.
B. The program cannot compile because the compiler cannot determine which max method should be invoked.
C. The program runs and prints 2 followed by "max(int, double)" is invoked.
D. The program runs and prints 2 followed by "max(double, int)" is invoked.
E. The program runs and prints "max(int, double) is invoked" followed by 2.

Your answer is correct
Explanation: This is known as ambiguous method invocation.

6.18 Analyze the following code.

```
public class Test {
  public static void main(String[] args) {
    System.out.println(m(2));
  }

  public static int m(int num) {
    return num;
  }

  public static void m(int num) {
    System.out.println(num);
  }
}
```

A. The program has a compile error because the two methods m have the same signature.
B. The program has a compile error because the second m method is defined, but not invoked in the main method.
C. The program runs and prints 2 once.
D. The program runs and prints 2 twice.

Your answer is correct
Explanation: You cannot override the methods based on the type returned.

Section 6.9 The Scope of Variables

6.19 A variable defined inside a method is referred to as _____.

A. a global variable
B. a method variable
C. a block variable
D. a local variable

Your answer is correct

6.20 What is k after the following block executes?

```
{
  int k = 2;
```

```
  nPrint("A message", k);
}
System.out.println(k);
```

- A. 0
- B. 1
- C. 2
- D. k is not defined outside the block. So, the program has a compile error

Your answer is correct

Explanation: k is defined inside the block. Outside the block, k is not defined.

Section 6.10 Case Study: Generating Random Characters

6.21 (int)(Math.random() * (65535 + 1)) returns a random number _____.

- A. between 1 and 65536
- B. between 1 and 65535
- C. between 0 and 65535
- D. between 0 and 65536

Your answer is correct

6.22 (int)('a' + Math.random() * ('z' - 'a' + 1)) returns a random number _____.

- A. between 0 and (int)'z'
- B. between (int)'a' and (int)'z'
- C. between 'a' and 'z'
- D. between 'a' and 'y'

Your answer is correct

6.23 (char)('a' + Math.random() * ('z' - 'a' + 1)) returns a random character _____.

- A. between 'a' and 'z'
- B. between 'a' and 'y'
- C. between 'b' and 'z'
- D. between 'b' and 'y'

Your answer is correct

6.24 Which of the following is the best for generating random integer 0 or 1?

- A. (int)Math.random()
- B. (int)Math.random() + 1
- C. (int)(Math.random() + 0.5)
- D. (int)(Math.random() + 0.2)
- E. (int)(Math.random() + 0.8)

Your answer is correct

Section 6.11 Method Abstraction and Stepwise Refinement

6.25 The client can use a method without knowing how it is implemented. The details of the implementation are encapsulated in the method and hidden from the client who invokes the method. This is known as _____.

- [x] A. information hiding
- [x] B. encapsulation
- [ ] C. method hiding
- [ ] D. simplifying method

Your answer is correct

6.26 _____ is to implement one method in the structure chart at a time from the top to the bottom.

- A. Bottom-up approach
- B. Top-down approach
- C. Bottom-up and top-down approach
- D. Stepwise refinement

Your answer is correct

6.27 _____ is a simple but incomplete version of a method.

- (●) A. A stub
- ( ) B. A main method
- ( ) C. A non-main method
- ( ) D. A method developed using top-down approach

Your answer is correct