Chapter 2
Working with Java Data Types

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

- ✓**Working with Java Data Types**
- Declare and initialize variables (including casting of primitive data types)
- Differentiate between object reference variables and primitive variables
- Know how to read or write to object fields
- Explain an Object's Lifecycle (creation, "dereference by reassignment" and garbage collection)
- Develop code that uses wrapper classes such as Boolean, Double, and Integer

1. Which of the following declarations does not compile?

1. `double num1, int num2 = 0;`
2. `int num1, num2;`
3. `int num1, num2 = 0;`
4. `int num1 = 0, num2 = 0;`

2. What is the output of the following?

```
public static void main(String... args) {
   String chair, table = "metal";
   chair = chair + table;
   System.out.println(chair);
}
```

1. `metal`
2. `metalmetal`
3. `nullmetal`
4. The code does not compile.

3. Which is correct about an instance variable of type `String`?

1. It defaults to an empty string.
2. It defaults to `null`.
3. It does not have a default value.
4. It will not compile without initializing on the declaration line.

4. Which of the following is not a valid variable name?

3. `blue$`
4. `Blue`

5. Which of these class names best follows standard Java naming conventions?

1. `fooBar`
2. `FooBar`
3. `FOO_BAR`
4. `F_o_o_B_a_r`

6. How many of the following methods compile?

```
public String convert(int value) {
   return value.toString();
}
public String convert(Integer value) {
   return value.toString();
}
public String convert(Object value) {
   return value.toString();
}
```

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

7. Which of the following does not compile?

1. `int num = 999;`
2. `int num = 9_9_9;`
3. `int num = _9_99;`
4. None of the above; they all compile.

8. Which of the following is a wrapper class?

1. `int`
2. `Int`
3. `Integer`
4. `Object`

9. What is the result of running this code?

```
public class Values {
   integer a = Integer.valueOf("1");
   public static void main(String[] nums) {
      integer a = Integer.valueOf("2");
      integer b = Integer.valueOf("3");
      System.out.println(a + b);
   }
}
```

1. `4`
2. `5`
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

10. Which best describes what the `new` keyword does?

1. Creates a copy of an existing object and treats it as a new one
2. Creates a new primitive
3. Instantiates a new object
4. Switches an object reference to a new one

11. Which is the first line to trigger a compiler error?

```
double d1 = 5f;    // p1
double d2 = 5.0;   // p2
```

```
float f1 = 5f;     // p3
float f2 = 5.0;    // p4
```

1. `p1`
2. `p2`
3. `p3`
4. `p4`

12. Which of the following lists of primitive types are presented in order from smallest to largest data type?

1. `byte`, `char`, `float`, `double`
2. `byte`, `char`, `double`, `float`
3. `char`, `byte`, `float`, `double`
4. `char`, `double`, `float`, `bigint`

13. Which of the following is not a valid order for elements in a class?

1. Constructor, instance variables, method names
2. Instance variables, constructor, method names
3. Method names, instance variables, constructor
4. None of the above: all orders are valid.

14. Which of the following lines contains a compiler error?

```
String title = "Weather";                // line x1
int hot, double cold;                    // line x2
System.out.println(hot + " " + title);   // line x3
```

1. `x1`
2. `x2`
3. `x3`
4. None of the above

15. How many instance initializers are in this code?

```
1:  public class Bowling {
2:     { System.out.println(); }
3:     public Bowling () {
4:        System.out.println();
5:     }
6:     static { System.out.println(); }
7:     { System.out.println(); }
8:  }
```

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

16. Of the types `double`, `int`, and `short`, how many could fill in the blank to have this code output `0`?

```
public static void main(String[] args) {
    _______defaultValue;
   System.out.println(defaultValue);
}
```

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

17. What is true of the `finalize()` method?

1. It may be called zero or one times.
2. It may be called zero or more times.
3. It will be called exactly once.

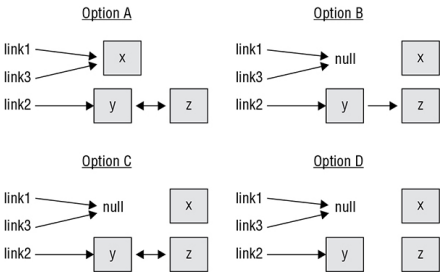
4. It may be called one or more times.

18. Which of the following is not a wrapper class?

1. `Double`
2. `Integer`
3. `Long`
4. `String`

19. Suppose you have the following code. Which of the images best represents the state of the references right before the end of the main method, assuming garbage collection hasn't run?

```
1:  public class Link {
2:     private String name;
3:     private Link next;
4:     public Link(String name, Link next) {
5:       this.name = name;
6:       this.next = next;
7:     }
8:     public void setNext(Link next) {
9:       this.next = next;
10:    }
11:    public Link getNext() {
12:       return next;
13:    }
14:    public static void main(String... args) {
15:       Link link1 = new Link("x", null);
16:       Link link2 = new Link("y", link1);
17:       Link link3 = new Link("z", link2);
18:       link2.setNext(link3);
19:       link3.setNext(link2);
20:       link1 = null;
21:       link3 = null;
22:    }
23: }
```



1. Option A
2. Option B
3. Option C
4. Option D

20. Which type can fill in the blank?

```
_______ pi = 3.14;
```

1. `byte`
2. `float`
3. `double`
4. `short`

21. What is the first line in the following code to not compile?

```
public static void main(String[] args) {
   int Integer = 0;        // k1
   Integer int = 0;        // k2
   Integer ++;             // k3
   int++;                  // k4
}
```

1. `k1`
2. `k2`
3. `k3`
4. `k4`

22. Suppose `foo` is a reference to an instance of a class. Which of the following is not true about `foo.bar`?

    1. `bar` is an instance variable.
    2. `bar` is a local variable.
    3. It can be used to read from `bar`.
    4. It can be used to write to `bar`.

23. Which of the following is not a valid class declaration?

    1. `class building {}`
    2. `class Cost$ {}`
    3. `class 5MainSt {}`
    4. `class _Outside {}`

24. Which of the following can fill in the blanks to make this code compile?

    ```
    _______d = new_______ (1_000_000_.00);
    ```

    1. `double`, `double`
    2. `double`, `Double`
    3. `Double`, `double`
    4. None of the above

25. Which is correct about a local variable of type `String`?

    1. It defaults to an empty string.
    2. It defaults to `null`.
    3. It does not have a default value.
    4. It will not compile without initializing on the declaration line.

26. Of the types `double`, `int`, `long`, and `short`, how many could fill in the blank to have this code output `0`?

    ```
    static  _______defaultValue;

       public static void main(String[] args) {
          System.out.println(defaultValue);
       }
    ```

    1. One
    2. Two
    3. Three
    4. Four

27. Which of the following is true about primitives?

    1. You can call methods on a primitive.
    2. You can convert a primitive to a wrapper class object simply by assigning it.
    3. You can convert a wrapper class object to a primitive by calling `valueOf()`.
    4. You can store a primitive directly into an `ArrayList`.

28. What is the output of the following?

    ```
    Integer integer = new Integer(4);
    System.out.print(integer.byteValue());

    System.out.print("-");

    int i = new Integer(4);
    System.out.print(i.byteValue());
    ```

    1. `4-0`
    2. `4-4`

3. The code does not compile.
4. The code compiles but throws an exception at runtime.

29. Given the following code, fill in the blank to have the code print `bounce`.

```
public class TennisBall {
   public TennisBall() {
      System.out.println("bounce");
   }
   public static void main(String[] slam) {
      ____________________
   }
}
```

1. `TennisBall;`
2. `TennisBall();`
3. `new TennisBall;`
4. `new TennisBall();`

30. Which of the following correctly assigns animal to both variables?

1. `String cat = "animal", dog = "animal";`
2. `String cat = "animal"; dog = "animal";`
3. `String cat, dog = "animal";`
4. `String cat, String dog = "animal";`

1. I
2. I, II
3. I, III
4. I, II, III, IV

31. Which two primitives have wrapper classes that are not merely the name of the primitive with an uppercase letter?

1. `byte` and `char`
2. `byte` and `int`
3. `char` and `int`
4. None of the above

32. Which of the following is true about `String` instance variables?

1. They can be set to `null`.
2. They can never be set from outside the class they are defined in.
3. They can only be set in the constructor.
4. They can only be set once per run of the program.

33. Which statement is true about primitives?

1. Primitive types begin with a lowercase letter.
2. Primitive types can be set to `null`.
3. `String` is a primitive.
4. You can create your own primitive types.

34. How do you force garbage collection to occur at a certain point?

1. Call `System.forceGc()`
2. Call `System.gc()`
3. Call `System.requireGc()`
4. None of the above

35. How many of the `String` objects are eligible for garbage collection right before the end of the `main` method?

```
public static void main(String[] fruits) {
   String fruit1 = new String("apple");
   String fruit2 = new String("orange");
   String fruit3 = new String("pear");
```

```
    fruit3 = fruit1;
    fruit2 = fruit3;
    fruit1 = fruit2;
  }
```

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

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

```
_______d = new_______ (1_000_000.00);
```

1. `double`, `double`
2. `double`, `Double`
3. `Double`, `double`
4. None of the above

37. What does the following output?

```
1:  public class InitOrder {
2:     public String first = "instance";
3:     public InitOrder() {
4:        first = "constructor";
5:     }
6:     { first = "block";  }
7:     public void print() {
8:        System.out.println(first);
9:     }
10:    public static void main(String... args) {
11:      new InitOrder().print();
12:    }
13: }
```

1. `block`
2. `constructor`
3. `instance`
4. The code does not compile.

38. How many of the following lines compile?

```
int i = null;
Integer in = null;
String s = null;
```

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

39. Which pairs of statements can accurately fill in the blanks in this table?

Variable Type	**Can be called within the class from what type of method**
Instance	Blank 1:_____
Static	Blank 2: _____

1. Blank 1: an instance method only, Blank 2: a static method only
2. Blank 1: an instance or static method, Blank 2: a static method only
3. Blank 1: an instance method only, Blank 2: an instance or static method
4. Blank 1: an instance or static method, Blank 2: an instance or static method

40. Which of the following does not compile?

1. `double num = 2.718;`
2. `double num = 2._718;`
3. `double num = 2.7_1_8;`
4. None of the above; they all compile.

41. Which of the following lists of primitive numeric types is presented in order from smallest to largest data type?

1. `byte`, `short`, `int`, `long`
2. `int`, `short`, `byte`, `long`
3. `short`, `byte`, `int`, `long`
4. `short`, `int`, `byte`, `long`

42. Fill in the blank to make the code compile:

```
package animal;
public class Cat {
  public String name;
  public static void main(String[] meow) {
    Cat cat = new Cat();
    _____________ = "Sadie";
  }
}
```

1. `cat.name`
2. `cat-name`
3. `cat.setName`
4. `cat[name]`

43. Which of the following is the output of this code, assuming it runs to completion?

```
package store;
public class Toy {
   public void play() {
      System.out.print("play-");
   }
   public void finalizer() {
      System.out.print("clean-");
   }
   public static void main(String[] fun) {
      Toy car = new Toy();
      car.play();
      System.gc();
      Toy doll = new Toy();
      doll.play();
   }
}
```

1. `play-`
2. `play-play-`
3. `play-clean-play-`
4. `play-play-clean-clean-`

44. Which is the most common way to fill in the blank to implement this method?

```
public class Penguin {

   private double beakLength;

   public static void setBeakLength(Penguin p, int b) {
      ____________________________
   }
}
```

1. `p.beakLength = b;`
2. `p['beakLength'] = b;`
3. `p[beakLength] = b;`
4. None of the above

45. Fill in the blanks to indicate whether a primitive or wrapper class can be assigned without the compiler using the autoboxing feature.

```
_______first = Integer.parseInt("5");
_______second = Integer.valueOf("5");
```

1. `int`, `int`
2. `int`, `Integer`
3. `Integer`, `int`
4. `Integer`, `Integer`

46. How many objects are eligible for garbage collection right before the end of the `main` method?

```
1:   public class Person {
2:      public Person youngestChild;
3:
4:      public static void main(String... args) {
5:         Person elena = new Person();
6:         Person diana = new Person();
7:         elena.youngestChild = diana;
8:         diana = null;
9:         Person zoe = new Person();
10:        elena.youngestChild = zoe;
11:        zoe = null;
12:     }
13:  }
```

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

47. Which is a valid constructor for this class?

```
public class TennisBall {
}
```

1. `public TennisBall static create() { return new TennisBall(); }`
2. `public TennisBall static newInstance() { return new TennisBall():}`
3. `public TennisBall() {}`
4. `public void TennisBall() {}`

48. Which of the following is not a possible output of this code, assuming it runs to completion?

```
package store;
public class Toy {
   public void play() {
      System.out.print("play-");
   }
   public void finalize() {
      System.out.print("clean-");
   }
   public static void main(String[] args) {
      Toy car = new Toy();
      car.play();
      System.gc();
      Toy doll = new Toy();
      doll.play();
   }
}
```

1. `play-`
2. `play-play-`
3. `play-play-clean-`
4. `play-play-clean-clean-`

49. Which converts a primitive to a wrapper class object without using autoboxing?

1. Call the `asObject()` method
2. Call the constructor of the wrapper class
3. Call the `convertToObject()` method
4. Call the `toObject()` method

50. What is the output of the following?

```
package beach;
public class Sand {
   public Sand() {
      System.out.print("a");
   }
   public void Sand() {
      System.out.print("b");
   }
   public void run() {
      new Sand();
      Sand();
   }
   public static void main(String... args) {
      new Sand().run();
   }
}
```

1. a
2. ab
3. aab
4. None of the above

PREV
Chapter 1 Java Basics

NEXT
Chapter 3 Using Operators and Decision Constructs