

## 1.9 Sample exam questions

Q1-1. Given:

```
class EJava {  
    //..code  
}
```

Which of the following options will compile?

- a**

```
package java.oca.associate;  
class Guru {  
    EJava eJava = new EJava();  
}
```
- b**

```
package java.oca;  
import EJava;  
class Guru {  
    EJava eJava;  
}
```
- c**

```
package java.oca.*;  
import java.default.*;  
class Guru {  
    EJava eJava;  
}
```

```
d package java.oqa.associate;
   import default.*;
   class Guru {
       default.EJava eJava;
   }
```

e None of the above

**Q1-2.** The following numbered list of Java class components is not in any particular order. Select the acceptable order of their occurrence in any Java class (choose all that apply):

- 1 comments
- 2 import statement
- 3 package statement
- 4 methods
- 5 class declaration
- 6 variables

- a 1, 3, 2, 5, 6, 4
- b 3, 1, 2, 5, 4, 6
- c 3, 2, 1, 4, 5, 6
- d 3, 2, 1, 5, 6, 4

**Q1-3.** Which of the following examples defines a correct Java class structure?

- a 

```
#connect java compiler;
#connect java virtual machine;
class EJavaGuru {}
```
- b 

```
package java compiler;
import java virtual machine;
class EJavaGuru {}
```
- c 

```
import javavirtualmachine.*;
package javacompiler;
class EJavaGuru {
    void method1() {}
    int count;
}
```
- d 

```
package javacompiler;
import javavirtualmachine.*;
class EJavaGuru {
    void method1() {}
    int count;
}
```
- e 

```
#package javacompiler;
$import javavirtualmachine;
class EJavaGuru {
    void method1() {}
    int count;
}
```

```
f package javacompiler;
   import javavirtualmachine;
   Class EJavaGuru {
       void method1() {}
       int count;
   }
```

**Q1-4.** Given the following contents of the Java source code file MyClass.java, select the correct options:

```
// contents of MyClass.java
package com.ejavaguru;
import java.util.Date;
class Student {}
class Course {}
```

- a The imported class, java.util.Date, can be accessed only in the class Student.
- b The imported class, java.util.Date, can be accessed by both the Student and Course classes.
- c Both of the classes Student and Course are defined in the package com.ejavaguru.
- d Only the class Student is defined in the package com.ejavaguru. The class Course is defined in the default Java package.

**Q1-5.** Given the following definition of the class EJavaGuru,

```
class EJavaGuru {
    public static void main(String[] args) {
        System.out.println(args[1]+":"+ args[2]+":"+ args[3]);
    }
}
```

what is the output of EJavaGuru, if it is executed using the following command?

```
java EJavaGuru one two three four
```

- a one:two:three
- b EJavaGuru:one:two
- c java:EJavaGuru:one
- d two:three:four

**Q1-6.** Which of the following options, when inserted at //INSERT CODE HERE, will print out EJavaGuru?

```
public class EJavaGuru {
    // INSERT CODE HERE
    {
        System.out.println("EJavaGuru");
    }
}
```

- a public void main (String[] args)
- b public void main(String args[])
- c static public void main (String[] array)
- d public static void main (String args)
- e static public main (String args[])

**Q1-7.** What is the meaning of “write once, run anywhere”? Select the correct options:

- a Java code can be written by one team member and executed by other team members.
- b It is for marketing purposes only.
- c It enables Java programs to be compiled once and can be executed by any JVM without recompilation.
- d Old Java code doesn’t need recompilation when newer versions of JVMs are released.

**Q1-8.** A class `Course` is defined in a package `com.ejavaguru`. Given that the physical location of the corresponding class file is `/mycode/com/ejavaguru/Course.class` and execution takes place within the `mycode` directory, which of the following lines of code, when inserted at `// INSERT CODE HERE`, will import the `Course` class into the class `MyCourse`?

```
// INSERT CODE HERE
class MyCourse {
    Course c;
}
```

- a import mycode.com.ejavaguru.Course;
- b import com.ejavaguru.Course;
- c import mycode.com.ejavaguru;
- d import com.ejavaguru;
- e import mycode.com.ejavaguru\*;
- f import com.ejavaguru\*;

**Q1-9.** Examine the following code:

```
class Course {
    String courseName;
}
class EJavaGuru {
    public static void main(String args[]) {
        Course c = new Course();
        c.courseName = "Java";
        System.out.println(c.courseName);
    }
}
```

Which of the following statements will be true if the variable `courseName` is defined as a private variable?

- a The class `EJavaGuru` will print Java.
- b The class `EJavaGuru` will print null.
- c The class `EJavaGuru` won't compile.
- d The class `EJavaGuru` will throw an exception at runtime.

**Q1-10.** Given the following definition of the class `Course`,

```
package com.ejavaguru.courses;
class Course {
    public String courseName;
}
```

what's the output of the following code?

```
package com.ejavaguru;
import com.ejavaguru.courses.Course;
class EJavaGuru {
    public static void main(String args[]) {
        Course c = new Course();
        c.courseName = "Java";
        System.out.println(c.courseName);
    }
}
```

- a The class `EJavaGuru` will print Java.
- b The class `EJavaGuru` will print null.
- c The class `EJavaGuru` won't compile.
- d The class `EJavaGuru` will throw an exception at runtime.

**Q1-11.** Given the following code, select the correct options:

```
package com.ejavaguru.courses;
class Course {
    public String courseName;
    public void setCourseName(private String name) {
        courseName = name;
    }
}
```

- a You can't define a method argument as a private variable.
- b A method argument should be defined with either `public` or default accessibility.
- c For overridden methods, method arguments should be defined with `protected` accessibility.
- d None of the above.

## **2.8    *Sample exam questions***

**Q2-1.** Given:

```
int myChar = 97;
int yourChar = 98;
System.out.print((char)myChar + (char)yourChar);

int age = 20;
System.out.print(" ");
System.out.print((float)age);
```

What is the output?

- a 195 20.0
- b 195 20
- c ab 20.0
- d ab 20
- e Compilation error
- f Runtime exception

**Q2-2.** Which of the options are correct for the following code?

```
public class Prim {                                // line 1
    public static void main(String[] args) {        // line 2
        char a = 'a';                               // line 3
        char b = -10;                               // line 4
        char c = '1';                               // line 5
        integer d = 1000;                           // line 6
        System.out.println(++a + b++ * c - d);      // line 7
    }                                                // line 8
}                                                  // line 9
```

- a Code at line 4 fails to compile.
- b Code at line 5 fails to compile.
- c Code at line 6 fails to compile.
- d Code at line 7 fails to compile.

**Q2-3.** What is the output of the following code?

```
public class Foo {
    public static void main(String[] args) {
        int a = 10;
        long b = 20;
        short c = 30;
        System.out.println(++a + b++ * c);
    }
}
```

- a 611
- b 641
- c 930
- d 960

**Q2-4.** Given:

```
Boolean buy = new Boolean(true);
Boolean sell = new Boolean(true);
System.out.print(buy == sell);
```

```
boolean buyPrim = buy.booleanValue();
System.out.print(!buyPrim);

System.out.print(buy && sell);
```

What is the output?

- a falsefalsefalse
- b truefalsetrue
- c falsetruetrue
- d falsefalsetrue
- e Compilation error
- f Runtime exception

**Q2-5.** Which of the following options contain correct code to declare and initialize variables to store whole numbers?

- a bit a = 0;
- b integer a2 = 7;
- c long a3 = 0x10C;
- d short a4 = 0512;
- e double a5 = 10;
- f byte a7 = -0;
- g long a8 = 123456789;

**Q2-6.** Select the options that, when inserted at // INSERT CODE HERE, will make the following code output a value of 11:

```
public class IncrementNum {
    public static void main(String[] args) {
        int ctr = 50;
        // INSERT CODE HERE
        System.out.println(ctr % 20);
    }
}
```

- a ctr += 1;
- b ctr == 1;
- c ++ctr;
- d ctr = 1;

**Q2-7.** What is the output of the following code?

```
int a = 10;
int b = 20;
int c = (a * (b + 2)) - 10 - 4 * ((2*2) - 6);
System.out.println(c);
```



- a 218
- b 232
- c 246
- d Compilation error

**Q2-8.** What is true about the following lines of code?

```
boolean b = false;
int i = 90;
System.out.println(i >= b);
```

- a Code prints true
- b Code prints false
- c Code prints 90 >= false
- d Compilation error

**Q2-9.** Examine the following code and select the correct options:

```
public class Prim {                                     // line 1
    public static void main(String[] args) {           // line 2
        int num1 = 12;                                 // line 3
        float num2 = 17.8f;                            // line 4
        boolean eJavaResult = true;                   // line 5
        boolean returnVal = num1 >= 12 && num2 < 4.567  // line 6
                               || eJavaResult == true;
        System.out.println(returnVal);                 // line 7
    }                                                    // line 8
}                                                        // line 9
```

- a Code prints false
- b Code prints true
- c Code will print true if code on line 6 is modified to the following:

```
boolean returnVal = (num1 >= 12 && num2 < 4.567) || eJavaResult == true;
```

- d Code will print true if code on line 6 is modified to the following:

```
boolean returnVal = num1 >= 12 && (num2 < 4.567 || eJavaResult == false);
```

**Q2-10.** Given:

```
boolean myBool = false;                                // line 1
int yourInt = 10;                                      // line 2
float hisFloat = 19.54f;                               // line 3
System.out.println(hisFloat = yourInt);                // line 4
System.out.println(yourInt > 10);                      // line 5
System.out.println(myBool = false);                   // line 6
```

What is the result?

- a** true  
true  
false
- b** 10.0  
false  
false
- c** false  
false  
false
- d** Compilation error

### 3.11 Sample exam questions

**Q3-1.** Which option defines a well-encapsulated class?

- a 

```
class Template {  
    public String font;  
}
```
- b 

```
class Template2 {  
    public String font;  
    public void setFont(String font) {  
        this.font = font;  
    }  
    public String getFont() {  
        return font;  
    }  
}
```
- c 

```
class Template3 {  
    private String font;  
    public String author;  
    public void setFont(String font) {  
        this.font = font;  
    }  
    public String getFont() {  
        return font;  
    }  
    public void setAuthor(String author) {  
        this.author = author;  
    }  
    public String getAuthor() {  
        return author;  
    }  
}
```
- d None of the above

**Q3-2.** Examine the following code and select the correct option(s):

```
public class Person {  
    public int height;  
    public void setHeight(int newHeight) {  
        if (newHeight <= 300)  
            height = newHeight;  
    }  
}
```

- a The height of a Person can never be set to more than 300.
- b The preceding code is an example of a well-encapsulated class.
- c The class would be better encapsulated if the height validation weren't set to 300.
- d Even though the class isn't well encapsulated, it can be inherited by other classes.

**Q3-3.** Which of the following methods correctly accepts three integers as method arguments and returns their sum as a floating-point number?

- a `public void addNumbers(byte arg1, int arg2, int arg3) {  
    double sum = arg1 + arg2 + arg3;  
}`
- b `public double subtractNumbers(byte arg1, int arg2, int arg3) {  
    double sum = arg1 + arg2 + arg3;  
    return sum;  
}`
- c `public double numbers(long arg1, byte arg2, double arg3) {  
    return arg1 + arg2 + arg3;  
}`
- d `public float wakaWakaAfrica(long a1, long a2, short a977) {  
    double sum = a1 + a2 + a977;  
    return (float)sum;  
}`

**Q3-4.** Which of the following statements are true?

- a If the return type of a method is `int`, the method can return a value of type `byte`.
- b A method may or may not return a value.
- c If the return type of a method is `void`, it can define a `return` statement without a value, as follows:

```
return;
```

- d A method may or may not accept any method arguments.
- e A method must accept at least one method argument or define its return type.
- f A method whose return type is `String` can't return `null`.

**Q3-5.** Given the following definition of class `Person`,

```
class Person {  
    public String name;  
    public int height;  
}
```

what is the output of the following code?

```
class EJavaGuruPassObjects1 {  
    public static void main(String args[]) {  
        Person p = new Person();  
        p.name = "EJava";  
        anotherMethod(p);  
        System.out.println(p.name);  
        someMethod(p);  
        System.out.println(p.name);  
    }  
}
```

```

static void someMethod(Person p) {
    p.name = "someMethod";
    System.out.println(p.name);
}
static void anotherMethod(Person p) {
    p = new Person();
    p.name = "anotherMethod";
    System.out.println(p.name);
}
}

```

- a** anotherMethod  
anotherMethod  
someMethod  
someMethod
- b** anotherMethod  
EJava  
someMethod  
someMethod
- c** anotherMethod  
EJava  
someMethod  
EJava
- d** Compilation error

**Q3-6.** What is the output of the following code?

```

class EJavaGuruPassPrim {
    public static void main(String args[]) {
        int ejg = 10;
        anotherMethod(ejg);
        System.out.println(ejg);
        someMethod(ejg);
        System.out.println(ejg);
    }
    static void someMethod(int val) {
        ++val;
        System.out.println(val);
    }
    static void anotherMethod(int val) {
        val = 20;
        System.out.println(val);
    }
}

```

- a** 20  
10  
11  
11
- b** 20  
20  
11  
10

- c 20  
10  
11  
10
- d Compilation error

**Q3-7.** Given the following signature of method `eJava`, choose the options that correctly overload this method:

```
public String eJava(int age, String name, double duration)

a private String eJava(int val, String firstName, double dur)
b public void eJava(int val1, String val2, double val3)
c String eJava(String name, int age, double duration)
d float eJava(double name, String age, byte duration)
e ArrayList<String> eJava()
f char[] eJava(double numbers)
g String eJava()
```

**Q3-8.** Given the following code,

```
class Course {
    void enroll(long duration) {
        System.out.println("long");
    }
    void enroll(int duration) {
        System.out.println("int");
    }
    void enroll(String s) {
        System.out.println("String");
    }
    void enroll(Object o) {
        System.out.println("Object");
    }
}
```

what is the output of the following code?

```
class EJavaGuru {
    public static void main(String args[]) {
        Course course = new Course();
        char c = 10;
        course.enroll(c);
        course.enroll("Object");
    }
}
```

- a Compilation error
- b Runtime exception

- c int  
String
- d long  
Object

**Q3-9.** Examine the following code and select the correct options:

```
class EJava {
    public EJava() {
        this(7);
        System.out.println("public");
    }
    private EJava(int val) {
        this("Sunday");
        System.out.println("private");
    }
    protected EJava(String val) {
        System.out.println("protected");
    }
}
class TestEJava {
    public static void main(String[] args) {
        EJava eJava = new EJava();
    }
}
```

- a The class EJava defines three overloaded constructors.
- b The class EJava defines two overloaded constructors. The private constructor isn't counted as an overloaded constructor.
- c Constructors with different access modifiers can't call each other.
- d The code prints the following:

```
protected
private
public
```

- e The code prints the following:

```
public
private
protected
```

**Q3-10.** Select the incorrect options:

- a If a user defines a private constructor for a public class, Java creates a public default constructor for the class.
- b A class that gets a default constructor doesn't have overloaded constructors.
- c A user can overload the default constructor of a class.
- d The following class is eligible for a default constructor:

```
class EJava {}
```

- e The following class is also eligible for a default constructor:

```
class EJava {  
    void EJava() {}  
}
```

**3.12**



## 4.9 Sample exam questions

**Q4-1.** What is the output of the following code?

```
class EJavaGuruArray {
    public static void main(String args[]) {
        int[] arr = new int[5];
        byte b = 4; char c = 'c'; long longVar = 10;
        arr[0] = b;
        arr[1] = c;
        arr[3] = longVar;
        System.out.println(arr[0] + arr[1] + arr[2] + arr[3]);
    }
}
```

- a 4c010
- b 4c10
- c 113
- d 103
- e Compilation error

**Q4-2.** What is the output of the following code?

```
class EJavaGuruArray2 {
    public static void main(String args[]) {
        int[] arr1;
        int[] arr2 = new int[3];
        char[] arr3 = {'a', 'b'};
        arr1 = arr2;
        arr1 = arr3;
        System.out.println(arr1[0] + ":" + arr1[1]);
    }
}
```

- a 0:0
- b a:b
- c 0:b
- d a:0
- e Compilation error

**Q4-3.** Which of the following are valid lines of code to define a multidimensional int array?

- a `int[][] array1 = {{1, 2, 3}, {}, {1, 2, 3, 4, 5}};`
- b `int[][] array2 = new array() {{1, 2, 3}, {}, {1, 2, 3, 4, 5}};`
- c `int[][] array3 = {1, 2, 3}, {0}, {1, 2, 3, 4, 5};`
- d `int[][] array4 = new int[2][];`

**Q4-4.** Which of the following statements are correct?

- a The following code executes without an error or exception:

```
ArrayList<Long> lst = new ArrayList<>();
lst.add(10);
```

- b Because ArrayList stores only objects, you can't pass an element of an ArrayList to a switch construct.
- c Calling `clear()` or `remove()` on an ArrayList will remove all its elements.
- d If you frequently add elements to an ArrayList, specifying a larger capacity will improve the code efficiency.
- e Calling the method `clone()` on an ArrayList creates its shallow copy; that is, it doesn't clone the individual list elements.

**Q4-5.** Which of the following statements are correct?

- a An ArrayList offers a resizable array, which is easily managed using the methods it provides. You can add and remove elements from an ArrayList.
- b Values stored by an ArrayList can be modified.
- c You can iterate through elements of an ArrayList using a for loop, Iterator, or ListIterator.
- d An ArrayList requires you to specify the total number of elements before you can store any elements in it.
- e An ArrayList can store any type of object.

**Q4-6.** What is the output of the following code?

```
import java.util.*;                                // line 1
class EJavaGuruArrayList {                          // line 2
    public static void main(String args[]) {         // line 3
        ArrayList<String> ejg = new ArrayList<>();   // line 4
        ejg.add("One");                              // line 5
        ejg.add("Two");                              // line 6
        System.out.println(ejg.contains(new String("One"))); // line 7
        System.out.println(ejg.indexOf("Two"));      // line 8
        ejg.clear();                                  // line 9
        System.out.println(ejg);                     // line 10
```

```

        System.out.println(ejg.get(1));           // line 11
    }                                             // line 12
}                                              // line 13

```

- a Line 7 prints true.
- b Line 7 prints false.
- c Line 8 prints -1.
- d Line 8 prints 1.
- e Line 9 removes all elements of the list ejg.
- f Line 9 sets the list ejg to null.
- g Line 10 prints null.
- h Line 10 prints [].
- i Line 10 prints a value similar to ArrayList@16356.
- j Line 11 throws an exception.
- k Line 11 prints null.

**Q4-7.** What is the output of the following code?

```

class EJavaGuruString {
    public static void main(String args[]) {
        String ejg1 = new String("E Java");
        String ejg2 = new String("E Java");
        String ejg3 = "E Java";
        String ejg4 = "E Java";
        do
            System.out.println(ejg1.equals(ejg2));
        while (ejg3 == ejg4);
    }
}

```

- a true printed once
- b false printed once
- c true printed in an infinite loop
- d false printed in an infinite loop

**Q4-8.** What is the output of the following code?

```

class EJavaGuruString2 {
    public static void main(String args[]) {
        String ejg = "game".replace('a', 'Z').trim().concat("Aa");
        ejg.substring(0, 2);
        System.out.println(ejg);
    }
}

```

- a gZmeAZ
- b gZmeAa

- c gZm
- d gZ
- e game

**Q4-9.** What is the output of the following code?

```
class EJavaGuruString2 {  
    public static void main(String args[]) {  
        String ejg = "game";  
        ejg.replace('a', 'Z').trim().concat("Aa");  
        ejg.substring(0, 2);  
        System.out.println(ejg);  
    }  
}
```

- a gZmeAZ
- b gZmeAa
- c gZm
- d gZ
- e game

**Q4-10.** What is the output of the following code?

```
class EJavaGuruStringBuilder {  
    public static void main(String args[]) {  
        StringBuilder ejg = new StringBuilder(10 + 2 + "SUN" + 4 + 5);  
        ejg.append(ejg.delete(3, 6));  
        System.out.println(ejg);  
    }  
}
```

- a 12S512S5
- b 12S12S
- c 1025102S
- d Runtime exception

**Q4-11.** What is the output of the following code?

```
class EJavaGuruStringBuilder2 {  
    public static void main(String args[]) {  
        StringBuilder sb1 = new StringBuilder("123456");  
        sb1.subSequence(2, 4);  
        sb1.deleteCharAt(3);  
        sb1.reverse();  
        System.out.println(sb1);  
    }  
}
```

- a 521
- b Runtime exception
- c 65321
- d 65431

**Q4-12.** What is the output of the following code?

```
String printDate = LocalDate.parse("2057-08-11")  
    .format(DateTimeFormatter.ISO_DATE_TIME);  
System.out.println(printDate);
```

- a August 11, 2057T00:00
- b Saturday Aug 11,2057T00:00
- c 08-11-2057T00:00:00
- d Compilation error
- e Runtime exception

## **5.10 Sample exam questions**

**Q5-1.** What's the output of the following code?

```
class Loop2 {  
    public static void main(String[] args) {  
        int i = 10;  
    }  
}
```

```
        do
            while (i < 15)
                i = i + 20;
            while (i < 2);
            System.out.println(i);
        }
    }
```

- a 10
- b 30
- c 31
- d 32

**Q5-2.** What's the output of the following code?

```
class Loop2 {
    public static void main(String[] args) {
        int i = 10;
        do
            while (i++ < 15)
                i = i + 20;
        while (i < 2);
        System.out.println(i);
    }
}
```

- a 10
- b 30
- c 31
- d 32

**Q5-3.** Which of the following statements is true?

- a The enhanced for loop can't be used within a regular for loop.
- b The enhanced for loop can't be used within a while loop.
- c The enhanced for loop can be used within a do-while loop.
- d The enhanced for loop can't be used within a switch construct.
- e All of the above statements are false.

**Q5-4.** What's the output of the following code?

```
int a = 10;
if (a++ > 10) {
    System.out.println("true");
}
{
    System.out.println("false");
}
System.out.println("ABC");
```

- a true  
false  
ABC
- b false  
ABC
- c true  
ABC
- d Compilation error

**Q5-5.** Given the following code, which of the optional lines of code can individually replace the `//INSERT CODE HERE` line so that the code compiles successfully?

```
class EJavaGuru {  
    public static void main(String args[]) {  
        int num = 10;  
        final int num2 = 20;  
        switch (num) {  
            // INSERT CODE HERE  
            break;  
            default: System.out.println("default");  
        }  
    }  
}
```

- a `case 10*3: System.out.println(2);`
- b `case num: System.out.println(3);`
- c `case 10/3: System.out.println(4);`
- d `case num2: System.out.println(5);`

**Q5-6.** What's the output of the following code?

```
class EJavaGuru {  
    public static void main(String args[]) {  
        int num = 20;  
        final int num2;  
        num2 = 20;  
        switch (num) {  
            default: System.out.println("default");  
            case num2: System.out.println(4);  
            break;  
        }  
    }  
}
```

- a default
- b default  
4
- c 4
- d Compilation error



**Q5-7.** What's the output of the following code?

```
class EJavaGuru {
    public static void main(String args[]) {
        int num = 120;
        switch (num) {
            default: System.out.println("default");
            case 0: System.out.println("case1");
            case 10*2-20: System.out.println("case2");
            break;
        }
    }
}
```

- a** default  
case1  
case2
- b** case1  
case2
- c** case2
- d** Compilation error
- e** Runtime exception

**Q5-8.** What's the output of the following code?

```
class EJavaGuru3 {
    public static void main(String args[]) {
        byte foo = 120;
        switch (foo) {
            default: System.out.println("ejavaguru"); break;
            case 2: System.out.println("e"); break;
            case 120: System.out.println("ejava");
            case 121: System.out.println("enum");
            case 127: System.out.println("guru"); break;
        }
    }
}
```

- a** ejava  
enum  
guru
- b** ejava
- c** ejavaguru  
e
- d** ejava  
enum  
guru  
ejavaguru

**Q5-9.** What's the output of the following code?

```
class EJavaGuru4 {  
    public static void main(String args[]) {  
        boolean myVal = false;  
        if (myVal=true)  
            for (int i = 0; i < 2; i++) System.out.println(i);  
        else System.out.println("else");  
    }  
}
```

- a** else
- b** 0  
1  
2
- c** 0  
1
- d** Compilation error

**Q5-10.** What's the output of the following code?

```
class EJavaGuru5 {  
    public static void main(String args[]) {  
        int i = 0;  
        for (; i < 2; i=i+5) {  
            if (i < 5) continue;  
            System.out.println(i);  
        }  
        System.out.println(i);  
    }  
}
```

- a** Compilation error
- b** 0  
5
- c** 0  
5  
10
- d** 10
- e** 0  
1  
5
- f** 5

## 6.10 *Sample exam questions*

**Q6-1.** What is the output of the following code?

```
class Animal {  
    void jump() { System.out.println("Animal"); }  
}  
class Cat extends Animal {  
    void jump(int a) { System.out.println("Cat"); }  
}  
class Rabbit extends Animal {  
    void jump() { System.out.println("Rabbit"); }  
}  
class Circus {  
    public static void main(String args[]) {  
        Animal cat = new Cat();  
        Rabbit rabbit = new Rabbit();  
        cat.jump();  
    }  
}
```

```
        rabbit.jump();  
    }  
}
```

- a Animal  
Rabbit
- b Cat  
Rabbit
- c Animal  
Animal
- d None of the above

**Q6-2.** Given the following code, select the correct statements:

```
class Flower {  
    public void fragrance() {System.out.println("Flower"); }  
}  
class Rose {  
    public void fragrance() {System.out.println("Rose"); }  
}  
class Lily {  
    public void fragrance() {System.out.println("Lily"); }  
}  
class Bouquet {  
    public void arrangeFlowers() {  
        Flower f1 = new Rose();  
        Flower f2 = new Lily();  
        f1.fragrance();  
    }  
}
```

- a The output of the code is  
Flower
- b The output of the code is  
Rose
- c The output of the code is  
Lily
- d The code fails to compile.

**Q6-3.** Examine the following code and select the correct method declaration to be inserted at //INSERT CODE HERE:

```
interface Movable {  
    void move();  
}
```

```

class Person implements Movable {
    public void move() { System.out.println("Person move"); }
}
class Vehicle implements Movable {
    public void move() { System.out.println("Vehicle move"); }
}
class Test {
    // INSERT CODE HERE
    movable.move();
}

```

- a void walk(Movable movable) {
- b void walk(Person movable) {
- c void walk(Vehicle movable) {
- d void walk() {

**Q6-4.** Select the correct statements:

- a Only an abstract class can be used as a base class to implement polymorphism with classes.
- b Polymorphic methods are also called overridden methods.
- c In polymorphism, depending on the exact type of object, the JVM executes the appropriate method at compile time.
- d None of the above.

**Q6-5.** Given the following code, select the correct statements:

```

class Person {}
class Employee extends Person {}
class Doctor extends Person {}

```

- a The code exhibits polymorphism with classes.
- b The code exhibits polymorphism with interfaces.
- c The code exhibits polymorphism with classes and interfaces.
- d None of the above.

**Q6-6.** Which of the following statements are true?

- a Inheritance enables you to reuse existing code.
- b Inheritance saves you from having to modify common code in multiple classes.
- c Polymorphism passes special instructions to the compiler so that the code can run on multiple platforms.
- d Polymorphic methods can't throw exceptions.

**Q6-7.** Given the following code, which of the options are true?

```
class Satellite {
    void orbit() {}
}
class Moon extends Satellite {
    void orbit() {}
}
class ArtificialSatellite extends Satellite {
    void orbit() {}
}
```

- a The method orbit defined in the classes Satellite, Moon, and Artificial-Satellite is polymorphic.
- b Only the method orbit defined in the classes Satellite and Artificial-Satellite is polymorphic.
- c Only the method orbit defined in the class ArtificialSatellite is polymorphic.
- d None of the above.

**Q6-8.** Examine the following code:

```
class Programmer {
    void print() {
        System.out.println("Programmer - Mala Gupta");
    }
}
class Author extends Programmer {
    void print() {
        System.out.println("Author - Mala Gupta");
    }
}
class TestEJava {
    Programmer a = new Programmer();
    // INSERT CODE HERE
    a.print();
    b.print();
}
```

Which of the following lines of code can be individually inserted at //INSERT CODE HERE so that the output of the code is as follows?

```
Programmer - Mala Gupta
Author - Mala Gupta
```

- a Programmer b = new Programmer();
- b Programmer b = new Author();
- c Author b = new Author();
- d Author b = new Programmer();

```
e Programmer b = ((Author)new Programmer());
f Author b = ((Author)new Programmer());
```

**Q6-9.** Given the following code, which of the options, when applied individually, will make it compile successfully?

```
Line1> interface Employee {}
Line2> interface Printable extends Employee {
Line3>     String print();
Line4> }
Line5> class Programmer {
Line6>     String print() { return("Programmer - Mala Gupta"); }
Line7> }
Line8> class Author extends Programmer implements Printable, Employee {
Line9>     String print() { return("Author - Mala Gupta"); }
Line10> }
```

- a Modify the code on line 2 to interface Printable{
- b Modify the code on line 3 to publicStringprint();
- c Define the accessibility of the print methods to public on lines 6 and 9.
- d Modify the code on line 8 so that it implements only the interface Printable.

**Q6-10.** What is the output of the following code?

```
class Base {
    String var = "EJava";
    void printVar() {
        System.out.println(var);
    }
}
class Derived extends Base {
    String var = "Guru";
    void printVar() {
        System.out.println(var);
    }
}
class QReference {
    public static void main(String[] args) {
        Base base = new Base();
        Base derived = new Derived();
        System.out.println(base.var);
        System.out.println(derived.var);
        base.printVar();
        derived.printVar();
    }
}
```

- a EJava  
EJava  
EJava  
Guru

- b** EJava  
Guru  
EJava  
Guru
- c** EJava  
EJava  
EJava  
EJava
- d** EJava  
Guru  
Guru  
Guru



## 7.8 Sample exam questions

**Q7-1.** What is the output of the following code:

```
class Course {
    String courseName;
    Course() {
        Course c = new Course();
        c.courseName = "Oracle";
    }
}

class EJavaGuruPrivate {
    public static void main(String args[]) {
        Course c = new Course();
        c.courseName = "Java";
        System.out.println(c.courseName);
    }
}
```

- a The code will print Java.
- b The code will print Oracle.
- c The code will not compile.
- d The code will throw an exception or an error at runtime.

**Q7-2.** Select the correct option(s):

- a You cannot handle runtime exceptions.
- b You should not handle errors.
- c If a method throws a checked exception, it must be either handled by the method or specified in its throws clause.
- d If a method throws a runtime exception, it may include the exception in its throws clause.
- e Runtime exceptions are checked exceptions.

**Q7-3.** Examine the following code and select the correct option(s):

```
class EJavaGuruExcep {
    public static void main(String args[]) {
        EJavaGuruExcep var = new EJavaGuruExcep();
        var.printArrValues(args);
    }
    void printArrValues(String[] arr) {
        try {
            System.out.println(arr[0] + ":" + arr[1]);
        } catch (NullPointerException e) {
            System.out.println("NullPointerException");
        } catch (IndexOutOfBoundsException e) {
            System.out.println("IndexOutOfBoundsException");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("ArrayIndexOutOfBoundsException");
        }
    }
}
```

```

    }
}

```

- a** If the class `EJavaGuruExcep` is executed using the following command, it prints `NullPointerException`:

```
java EJavaGuruExcep
```

- b** If the class `EJavaGuruExcep` is executed using the following command, it prints `IndexOutOfBoundsException`:

```
java EJavaGuruExcep
```

- c** If the class `EJavaGuruExcep` is executed using the following command, it prints `ArrayIndexOutOfBoundsException`:

```
java EJavaGuruExcep one
```

- d** The code will fail to compile.

**Q7-4.** What is the output of the following code?

```

class EJava {
    void method() {
        try {
            guru();
            return;
        } finally {
            System.out.println("finally 1");
        }
    }
    void guru() {
        System.out.println("guru");
        throw new StackOverflowError();
    }
    public static void main(String args[]) {
        EJava var = new EJava();
        var.method();
    }
}

```

- a** guru  
finally 1
- b** guru  
finally 1  
Exception in thread "main" java.lang.StackOverflowError
- c** guru  
Exception in thread "main" java.lang.StackOverflowError
- d** guru
- e** The code fails to compile.

**Q 7-5.** What is the output of the following code?

```
class Quest5 {
    public static void main(String args[]) {
        int arr[] = new int[5];
        arr = new int[]{1,2,3,4};

        int x = arr[1]-- + arr[0]-- /arr[0] * arr[4];
        System.out.println(x);
    }
}
```

- a The code outputs a value.
- b The code outputs a value followed by an exception.
- c `ArithmeticException`
- d `NullPointerException`
- e `IndexOutOfBoundsException`
- f `ArrayIndexOutOfBoundsException`
- g Compilation error
- h None of the above

**Q7-6.** Which of the following methods will not compile?

- a 

```
private void method1(String name) {
    if (name.equals("star"))
        throw new IllegalArgumentException(name);
}
```
- b 

```
private void method2(int age) {
    if (age > 30)
        throw Exception();
}
```
- c 

```
public Object method3(boolean accept) {
    if (accept)
        throw new StackOverflowError();
    else
        return new StackOverflowError();
}
```
- d 

```
protected double method4() throws Exception {
    throw new Throwable();
}
```
- e 

```
public double method5() throws Exception {
    return 0.7;
}
```

**Q7-7.** What is the output of the following code?

```
class TryFinally {
    int tryAgain() {
        int a = 10;
        try {
            ++a;
        }
    }
}
```

```

        } finally {
            a++;
        }
        return a;
    }
    public static void main(String args[]) {
        System.out.println(new TryFinally().tryAgain());
    }
}

```

- a 10
- b 11
- c 12
- d Compilation error
- e Runtime exception

**Q7-8.** What is the output of the following code?

```

class EJavaBase {
    void myMethod() throws ExceptionInInitializerError {
        System.out.println("Base");
    }
}
class EJavaDerived extends EJavaBase {
    void myMethod() throws RuntimeException {
        System.out.println("Derived");
    }
}
class EJava3 {
    public static void main(String args[]) {
        EJavaBase obj = new EJavaDerived();
        obj.myMethod();
    }
}

```

- a Base
- b Derived
- c Derived  
Base
- d Base  
Derived
- e Compilation error

**Q7-9.** Which of the following statements are true?

- a A user-defined class may not throw an `IllegalStateException`. It must be thrown only by Java API classes.
- b `System.out.println` will throw a `NullPointerException` if an uninitialized instance variable of type `String` is passed to it to print its value.

- c NumberFormatException is thrown by multiple methods from the Java API when invalid numbers are passed on as Strings to be converted to the specified number format.
- d ExceptionInInitializerError may be thrown by the JVM when a static initializer in your code throws a NullPointerException.

**Q7-10.** What is the output of the following code?

```
class EJava {  
    void foo() {  
        try {  
            String s = null;  
            System.out.println("1");  
            try {  
                System.out.println(s.length());  
            } catch (NullPointerException e) {  
                System.out.println("inner");  
            }  
            System.out.println("2");  
        } catch (NullPointerException e) {  
            System.out.println("outer");  
        }  
    }  
    public static void main(String args[]) {  
        EJava obj = new EJava();  
        obj.foo();  
    }  
}
```

- a 1  
inner  
2  
outer
- b 1  
outer
- c 1  
inner
- d 1  
inner  
2