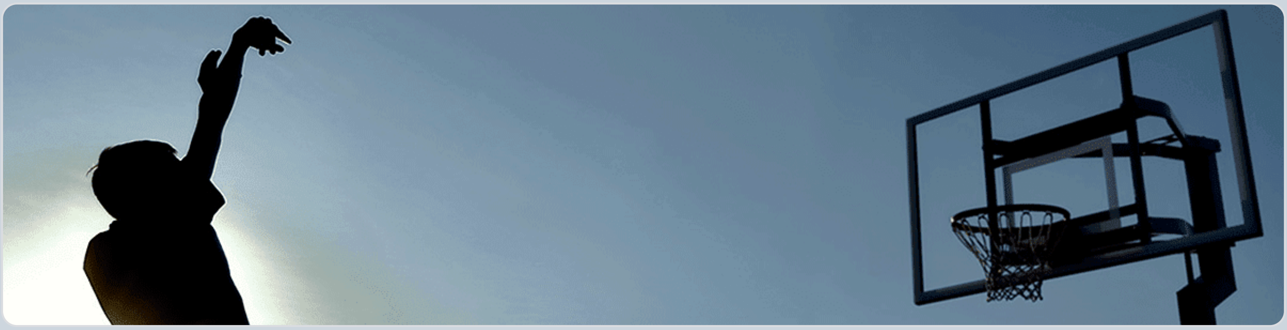


Oracle Certified Professional Java SE11 Developer

Total points 51/60

Email *

bhawukarora042@gmail.com

✓

```
public class test2
{
        public static String Computing(Object Process)
        {
                return "3"+Process.toString();
        }

         public static void main(String[] a)
        {
                System.out.println(Computing(5*5));
        }
}
```

* 3/3

○ 35*5

○ 28

● 325 ✓

○ does not compile

○ run time exception

✓ * 3/3

```
public class test2
{
      public String cloud(int vm)
      {
            return "1";
      }

      public String cloud(Integer vm)
      {
            return "2"+vm.toString();
      }

       public static void main(String[] a)
      {
            System.out.println(cloud((short)5));
      }
}
```

○ 1

○ 2

● does not compile ✓

○ run time exception

✓

```
public class test2
{
        private Integer eggs = 2;
        {
                this.eggs = 3;
        }

        public test2(int eggs)
        {
                this.eggs = eggs;
        }

        public static void main(String[] r)
        {
                var c1 = new test2(1);
                var c2 = new test2(2);
                var c3 = new test2(3);
                c1.eggs = c2.eggs;
                c2 = c1;
                c3.eggs = null;
                System.out.println(c1.eggs +""+c2.eggs+""+c3.eggs );
        }
}
```

* 3/3

● 22null ✓

○ 12null

○ 123

○ does not compile

○ exception

✗ *0/3

```
import java.time.*;
import java.time.format.*;
public class Test
{
      public static void main(String[] args)
      {
            LocalDateTime pi = LocalDateTime.of(2023, 3, 14, 1, 59);
            DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd mm
yyyy a");
            System.out.print(formatter.format(pi));
      }
}
```

● 14 03 2023 AM ✗

○ 14 59 2023 AM

○ 1 03 2023 AM

○ 1 03 2023 PM

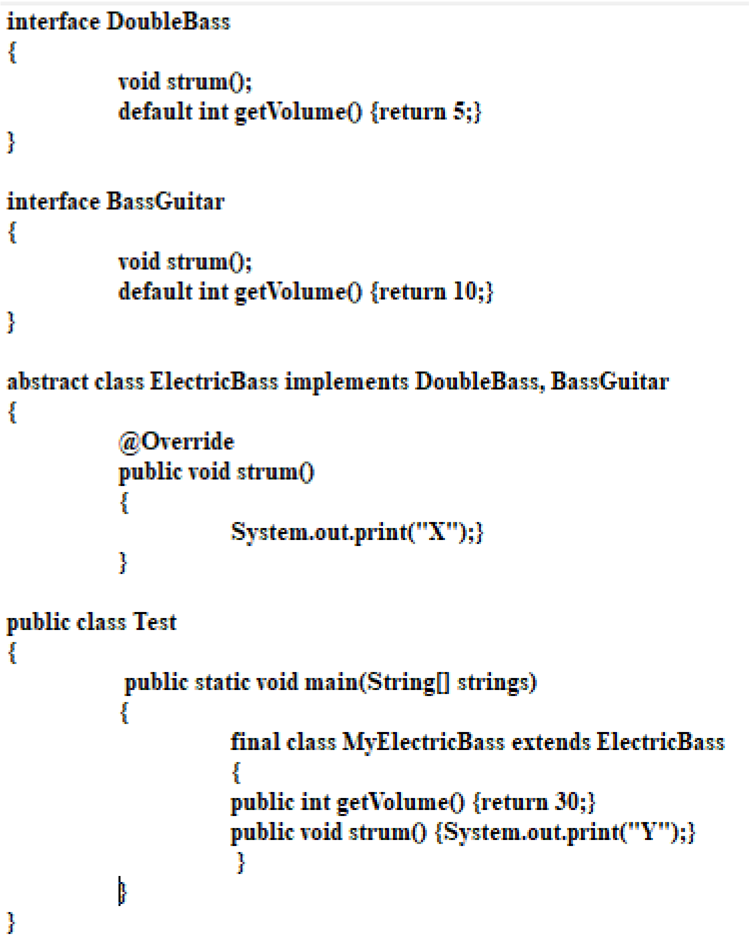
○ Option 5

Correct answer

● 14 59 2023 AM

✓ Output *

3/3



```
interface DoubleBass
{
        void strum();
        default int getVolume() {return 5;}
}

interface BassGuitar
{
        void strum();
        default int getVolume() {return 10;}
}

abstract class ElectricBass implements DoubleBass, BassGuitar
{
        @Override
        public void strum()
        {
                System.out.print("X");}
        }

public class Test
{
         public static void main(String[] strings)
        {
                final class MyElectricBass extends ElectricBass
                {
                public int getVolume() {return 30;}
                public void strum() {System.out.print("Y");}
                 }
        }
}
```

○ X

○ Y

○ X Y

● Compile time error ✓

○ Runtime error

✓ 2/2

```
final class Oracle *
{
        void Java(int chapter) throws Exception
        {
        }
}

public class test2 extends Oracle
{
        final void Java(final int chapter)
        {
                switch(chapter)
                {
                case 2: System.out.print(9);
                default: System.out.print(3);
                }
        }

        public static void main(String... u)
        {
                var v = new test2();
                v.Java(2);
        }
}
```

○ 93

○ 9

● does not compile ✓

○ exception

✓
```
public class test2
{
      int US = 100;
      public static void main(String[] data)
      {
            var IN = 500;
            System.out.print(US+IN);
}
}
```
* 3/3

○ 600

○ 500

○ 100 + 500

● does not compile ✓

○ exception

✓ * 3/3

```
public class Test
{
    public void doTaxes() throws Throwable
    {
        try
        {
            throw new NumberFormatException();
        } catch (ClassCastException | ArithmeticException e)
        {
            System.out.println("Math");
        } catch (Exception f)
        {
            System.out.println("Unknown");
        }
    }

    public static void main(String[] numbers) throws Throwable
    {
        try
        {
            new Test().doTaxes();
        } finally
        {
            System.out.println("Done!");
        }
    }
}
```

○ Math

○ Math Done

○ Unknown

● Unknown Done ✓

○ Compile time error

○ Runtime error

✓

```
public class test2
{
        static int Java=11;
        static
        {
                Java++;
        }
        private static void drive()
        {
                 Java++;
                {
                        Java++;
                }
        }
        public static void main(String[] args)
        {
                drive();
                drive();
                System.out.println(Java);
        }
}
```

Output? *

3/3

○ 11

● 16 ✓

○ 14

○ does not compile

○ exception

✓

```
interface India
{
    void UP(double psi);
}
interface Nepal extends India
{
    void Kathmandu(double tika);
}

public class test2
{
    public static final void add(Nepal in, double input)
    {
    in.Kathmandu(input);
    }

    public static void main(String... future)
    {
        final test2 r = new test2();
        r.add (x-> System.out.print(x+" Kathmandu!"), 5);
    }
}
```

* 3/3

● does not compile ✓

○ exception

○ 5

○ Kathmandu 5

✓ * 3/3

```
enum Google
{
     Gmail, Drive, Forms;
}

public class test2
{
     public static void main(String... unused)
     {
      final Google input = Google.Youtube;

     switch(input)
     {
          case Gmail:
                    System.out.print("1");
                    break;
          case Drive:
                    System.out.print("2");
                    break;
          case Forms:
                    System.out.print("3");
                    break;
     }
     }
}
```

● does not compile ✓

○ 1

○ 2

○ 3

○ run time exception

✓ * 3/3

```
interface Super
{
        default int shock()
        { return 7; }
}

interface Spider
{
        default int shock()
        { return 5; } }

public class test2 implements Spider, Super
{
        public int shock(String... x)
        {
                return x.length;
        }

        public static void main(String[] notes)
        {
                 System.out.print(new test2().shock());
        }
}
```

● does not compile ✓

○ 7

○ 5

○ run time exception

○ null

✗ module com.apple { exports com.apple; } * 0/3

module com.4apple { requires com.apple;}

module com.apple4 { declares com.apple; }

module com.apple-four { }

module com.apple$ {}

How many of these module declarations are valid?

○ None

○ One

○ Two

● Three ✗

○ Four

○ Five

Correct answer

● Two

✓ * 3/3

```
public class Test
{
    interface Target
    {
        boolean needToAim(double angle);
    }

    static void prepare(double angle, Target t)
    {
        boolean ready = t.needToAim(angle); // k1
        System.out.println(ready);
    }

    public static void main(String[] args)
    {
        prepare(45, d => d > 5 || d < -5); // k2
     }
}
```

○ true

○ false

● Compilation error ✓

○ Runtime exception

✓ interface Go * 3/3

```
interface Go
{
        int sell();
}

abstract class Went implements Go
{
        public int sell()
        {
                return 5;
        }
}

public class test2 extends Went
{
        public final int sell()
        {
                return "3";
        }

        public static void main(String[] expensive)
        {
                System.out.print(new test2().sell());
        }
}
```

○ 3

○ 5

● does not compile ✓

○ run time exception

✓ * 3/3

```
import java.util.*;
public class Test
{
        public static void main(String[] args)
        {
                args = new String[] {"0", "1", "01", "10" };
                Arrays.sort(args);
                System.out.println(Arrays.toString(args));
        }
}
```

○ Compile time error

○ 0 1 01 10

● 0 01 1 10 ✓

○ 10 1 01 0

○ 1 10 01 0

○ Runtime error

✓ * 4/4

```
interface abc
{
        static void hello() //L1
        {
                System.out.println("Hello");
        }
}

class ghj
{
        static void hi() //L2
        {
                System.out.println("Hi");
        }
}

class Test extends ghj implements abc
{
        public static void main(String ar[])
        {
        Test.hi(); //L3
        Test.hello(); //L4
        }
}
```

○ Compilation error at L1

○ Compilation error at L2

○ Compilation error at L3

● Compilation error at L4 ✓

○ No Compilation error

○ Runtime Exception

✓

```
public class test2
{
        private int size;
        // insert constructor here

        public static void sendHome(test2 p, int newSize)
        {
                 p = new test2(newSize);
                p.size = 4;
        }

         public static final void main(String... params)
        {
                final var t = new test2(3);
                sendHome(t,7);
                System.out.print(t.size);
        }
}
```

* 3/3

● public test2(int size) {this.size=size;} ✓

○ public void test2(int size) {this.size=size;}

○ public static test2 newInstance(int size) {this.size=size;}

○ public static test2 create(int size) {this.size=size;}

✓ *   3/3

```
public class test2
{
        class abc extends test2 {}

        public void roar()
        {
                var t1 = new test2();
                //
                //



        }
}
```

○ t1.abc();

○ test2.new abc();

● t1.new abc(); ✓

○ new test2.abc();

✗ 1. --module-path and -classpath *

2. --module-path and -class-path

3. --module-path and --class-path

4. --path and -classpath

5. --path and -class-path

6. --path and --class-path

Select the correct one...

0/3

● 1 and 2 ✗

○ 2 and 3

○ 3 and 1

○ 3 and 4

○ 4 and 6

○ 5 and 1

○ 6 and 2

Correct answer

● 3 and 1