# Oracle.1z0-808.v2019-08-30.q89

| | |
|---|---|
| **Exam Code:** | 1z0-808 |
| **Exam Name:** | Java SE 8 Programmer I |
| **Certification Provider:** | Oracle |
| **Free Question Number:** | 89 |
| **Version:** | v2019-08-30 |
| **# of views:** | 1081 |
| **# of Questions views:** | 23886 |
| https://www.freecram.com/torrent/Oracle.1z0-808.v2019-08-30.q89.html | |

**NEW QUESTION: 1**
Which two statements are true? (Choose two.)
**A.** Error is a Throwable.
**B.** Error class is unextendable.
**C.** Error is a RuntimeException.
**D.** Error class is extendable.
**E.** Error is an Exception.
**Answer: C,D (LEAVE A REPLY)**

**NEW QUESTION: 2**
Given the code fragments:

Person.java:

```
public class Person {
    String name;
    int age;

    public Person(String n, int a) {
        name = n;
        age = a;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {
    for (Person p : list) {
        if (predicate.test(p)) {
            System.out.println(p.name + " ");
        }
    }
}

public static void main(String[] args) {
    List<Person> iList = Arrays.asList(new Person("Hank", 45),
                                       new Person("Charlie", 40),
                                       new Person("Smith", 38));
    //line n1
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

A:

```
checkAge (iList, ( ) -> p. get Age ( ) > 40);
```

B:

```
checkAge(iList, Person p -> p.getAge( ) > 40);
```

C:

```
checkAge (iList, p -> p.getAge ( ) > 40);
```

D:

```
checkAge(iList, (Person p) -> { p.getAge() > 40; });
```

**A.** Option D

**B.** Option B

**C.** Option C

**D.** Option A

**Answer: C (LEAVE A REPLY)**

**NEW QUESTION: 3**

Given:

```
class Test {
   public static void main (String [] args) {
         int numbers [ ];
         numbers = new int [2];
         numbers [0] = 10;
         numbers [1] = 20;

         numbers = new int [4];
         numbers [2] = 30;
         numbers [3] = 40;
         for (int x : numbers) {
             System.out.print (" " + x) ;
         }
   }
}
```



What is the result?

**A.** An exception is thrown at runtime.

**B.** Compilation fails.

**C.** 10 20 30 40

**D.** 0 0 30 40

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 4**

Given:

```
class Vehicle {
    int x;
    Vehicle(){
        this(10);  // line n1
    }
    Vehicle(int x) {
        this.x = x;
    }
}

class Car extends Vehicle {
    int y;
    Car() {
        super(10);   // line n2
    }
    Car(int y) {
       super(y);
       this.y = y;
    }
    public String toString() {
        return super.x + ":" + this.y;
    }
}
```

And given the code fragment:

```
Vehicle y = new Car(20);
System.out.println(y);
```

What is the result?

**A.** Compilation fails at line n1.

**B.** 20:20

**C.** Compilation fails at line n2.

**D.** 10:20

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 5**

Given the code fragment:

```
int wd = 0;
String days[] = {"sun", "mon", "wed", "sat"};
for (String s:days) {
    switch (s) {
        case "sat":
        case "sun":
            wd -= 1;
            break;
        case "mon":
            wd -= 1;
            break;
        case "wed":
            wd += 2;
    }
}
```

What is the result?

**A.** 3

**B.** -1

**C.** Compilation fails.

**D.** 0

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 6**

Which two statements are true about Java byte code? (Choose two.)

**A.** It has ".java" extension.

**B.** It can run on any platform.

**C.** It can run on any platform that has a Java compiler.

**D.** It can be serialized across network.

**E.** It can run on any platform that has the Java Runtime Environment.

**Answer: D,E (LEAVE A REPLY)**

**NEW QUESTION: 7**

Given:

```
public class Test {
    public static final int MIN =1;
    public static void main (String [] args) {
        int x = args.length;
        if (checkLimit (x)) {    //line n1
            System,out.println ("Java SE");
        } else {
            System.out.println ("Java EE");
        }
    }
    public static boolean checkLimit (int x) {
        return (x > = MIN) ? true : false;
    }
}
```



And given the commands:

```
javac Test.java
java Test
```



What is the result?

**A.** Java EE

**B.** A NullPointerException is thrown at runtime.

**C.** Java SE

**D.** Compilation fails at line n1.

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 8**

Given:

```
public class Test {
    int x, y;

    public Test(int x, int y) {
        initialize(x, y);
    }

    public void initialize(int x, int y) {
        this.x = x * x;
        this.y = y * y;
    }

    public static void main(String[] args) {
        int x = 9, y = 5;
        Test obj = new Test(x, y);
        System.out.println(x + " " + y);
    }
}
```

What is the result?

**A.** 9 5

**B.** 81 25

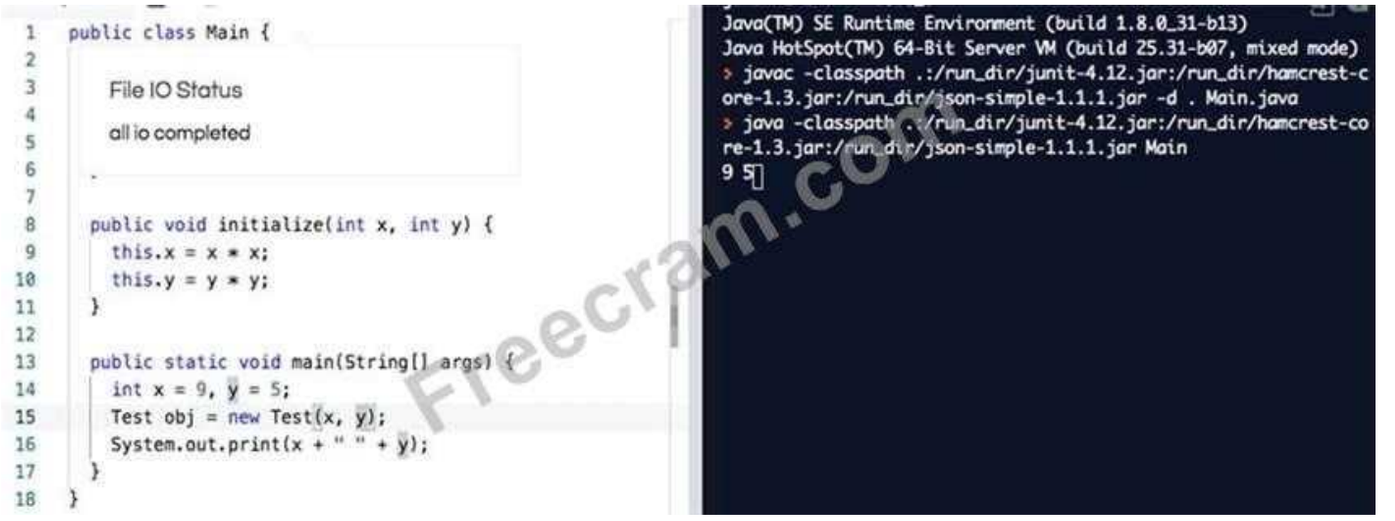
**C.** Compilation fails.

**D.** 0 0

**Answer: A (LEAVE A REPLY)**

Explanation/Reference:

Explanation:




**NEW QUESTION: 9**

Which statement will empty the contents of a StringBuilder variable named sb?

**A.** sb. delete (0, sb. length () );

**B.** sb. delete (0, sb. size () );

**C.** sb. removeAll ();

**D.** sb. deleteAll ();

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 10**

Given:

```
public class MyClass {
    public static void main(String[] args) {
        String s = "Java Duke";
        int len = s.trim().length();
        System.out.print(len);
    }
}
```

What is the result?

**A.** 8

**B.** 10

**C.** Compilation fails.

**D.** 11

**E.** 9

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 11**

Given the code fragment:

```
Public static void main (String [] args) {
    System.out.println ("Result A " + 0 + 1);
    System.out.println ("Result B " + (1) + (2) );
}
```

What is the result?

A:

```
Result A 01
Result B 3
```

B:

```
Result A 1
Result B 12
```

C:

```
Result A 1
Result B 3
```

D:

```
Result A 01
Result B 12
```

**A.** Option B

**B.** Option D

**C.** Option C

**D.** Option A

**Answer: C (LEAVE A REPLY)**

**NEW QUESTION: 12**

Given:

```
class CD {
    int r;
    CD(int r){
        this.r=r;
    }
}

class DVD extends CD {
    int c;
    DVD(int r, int c) {
        // line n1
    }
}
```

And given the code fragment:

```
DVD dvd = new DVD(10,20);
```

Which code fragment should you use at line n1 to instantiate the dvd object successfully?

A)
```
super.r = r;
this.c = c;
```

B)
```
super(r);
this(c);
```

C)
```
super(r);
this.c = c;
```

D)
```
this.c = r;
super(c);
```

**A.** Option B

**B.** Option D

**C.** Option A

**D.** Option C

**Answer: D (LEAVE A REPLY)**

**NEW QUESTION: 13**

Given the code fragment:

```
public static void main (String [] args) {
    String names [] = ("Thomas", "Peter", "Joseph");
    String pwd [] = new String [3];
    int idx = 0;
    try {
        for (String n: names) {
            pwd [idx] = n.substring (2, 6);
            idx++;
        }
    }
    catch (Exception e) {
        System.out.println ("Invalid Name");
    }
    for (String p: pwd) {
        System.out.println (p);
    }
}
```

What is the result?

**A.** Invalid Name

```
Invalid Name
 omas
 null
```

**B.**

```
null
omas
 ter
 seph
```

**C.**

```
Invalid Name
 omas
```

**D.**

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 14**

Given the code fragment:

```
int nums1[] = new int[3];
int nums2[] = {1, 2, 3, 4, 5};
nums1 = nums2;
for (int x : nums1){
    System.out.print(x + ":");
}
```

What is the result?

**A.** An ArrayoutofBoundsException is thrown at runtime.

**B.** Compilation fails.

**C.** 1:2:3:

**D.** 1:2:3:4:5:

**Answer: D (LEAVE A REPLY)**

**NEW QUESTION: 15**

Given:

```
public class Test {

    public static void main(String[] args) {

        String[][] chs = new String[2][];
        chs[0] = new String[2];
        chs[1] = new String[5];
        int i = 97;

        for (int a = 0; a < chs.length; a++) {
            for (int b = 0; b < chs.length; b++) {
                chs[a][b] = "" + i;
                i++;
            }
        }

        for (String[] ca : chs) {
            for (String c : ca) {
                System.out.print(c + " ");
            }
            System.out.println();
        }
    }
}
```



What is the result?

**A.** An ArrayIndexOutOfBoundsException is thrown at runtime.

**B.** Compilation fails.

**C.** 97 98
99 100 null null null

**D.** A NullPointerException is thrown at runtime.

**E.** 97 98
99 100 101 102 103

**Answer: C (LEAVE A REPLY)**

**NEW QUESTION: 16**

Given the code snippet from a compiled Java source file:

```
public class MyFile
{
      public static void main (String[] args)
      {
            String arg1 = args[1];
            String arg2 = args[2];
            String arg3 = args[3];
            System.out.println("Arg is " + arg3);
      }
}
```

Which command-line arguments should you pass to the program to obtain the following output?
Arg is 2

**A.** java MyFile 0 1 2 3
**B.** java MyFile 1 3 2 2
**C.** java MyFile 2 2 2
**D.** java MyFile 1 2 2 3 4
**Answer: B (LEAVE A REPLY)**



**NEW QUESTION: 17**
Given:

```
public class MyField {
    int x;
    int y;
    public void doStuff(int x, int y) {
        this.x = x;
        y = this.y;
    }
    public void display () {
        System.out.print(x + " " + y + " : ");
    }
    public static void main(String[] args) {
        MyField m1 = new MyField();
        m1.x = 100;
        m1.y = 200;
        MyField m2 = new MyField();
        m2.doStuff(m1.x, m1.y);
        m1.display();
        m2.display();
    }
}
```



What is the result?

**A.** 100 200 : 100 0 :

**B.** 100 0 : 100 0 :

**C.** 100 0 : 100 200:

**D.** 100 200 : 100 200 :

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 18**

Given the code fragments:

```
interface Exportable {
     void export();
}

class Tool implements Exportable {
    public void export() {                    // line n1
        System.out.println("Tool::export");
    }
}

class ReportTool extends Tool {

    void export() {                           // line n2
        System.out.println("RTool::export");
    }

    public static void main(String[] args) {
        Tool aTool = new ReportTool();
        Tool bTool = new Tool();
        callExport(aTool);
        callExport(bTool);
    }

    public static void callExport(Exportable ex)  {
        ex.export();
    }
}
```



What is the result?

**A.** Compilation fails only at line n1.

**B.** Tool::export
Tool::export

**C.** Compilation fails at both line n1 and line2.

**D.** Compilation fails only at line n2.

**E.** RTool::export
Tool::export

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 19**

Which one of the following code examples uses valid Java syntax?

```
A.
public class Boat {

        public static void main (String [] args) {
                System.out.println  ("I float.");
                }
}

B.
public class Cake {
         public static void main (String [] ) {
                System.out.println  ("Chocolate");
                }
}

C.
public class Dog {
         public void main (String [] args) {
                System.out.println  ("Squirrel.");
                }
}

D.
public class Bank {
         public static void main (String () args) {
                System.out.println  ("Earn interest.");
                }
}
```



**A.** Option A
**B.** Option B
**C.** Option C
**D.** Option D
**Answer: A (LEAVE A REPLY)**
Explanation/Reference:
Reference: https://docs.oracle.com/javase/tutorial/getStarted/application/

**NEW QUESTION: 20**
Given the code fragment:

```
abstract class Planet {
    protected void revolve() {            //line n1
    }

    abstract void rotate();               //line n2
}

class Earth extends Planet {
    void revolve() {                      //line n3
    }

    protected void rotate() {             //line n4
    }
}
```

Which two modifications, made independently, enable the code to compile? (Choose two.)

**A.** Make the method at line n3 public.

**B.** Make the method at line n4 public.

**C.** Make the method at line n1 public.

**D.** Make the method at line n3 protected.

**E.** Make the method at line n2 public.

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 21**

Given the code fragment:

```
public static void main(String[] args) {
    StringBuilder sb = new StringBuilder(5);
    String s = "";

    if (sb.equals(s)) {
        System.out.println("Match 1");
    } else if (sb.toString().equals(s.toString())) {
        System.out.println("Match 2");
    } else {
        System.out.println("No Match");
    }
}
```

What is the result?

**A.** A NullPointerException is thrown at runtime.

**B.** Match 1

**C.** No Match

**D.** Match 2

**Answer: D (LEAVE A REPLY)**

**NEW QUESTION: 22**

Given the code fragment:

```
24. float var1 = (12_345.01 >= 123_45.00) ? 12_456 : 124_56.02f;
25. float var2 = var1 + 1024;
26. System.out.print(var2);
```

What is the result?

**A.** 13480.02

**B.** 13480.0

**C.** Compilation fails.

**D.** An exception is thrown at runtime.

**Answer: B (LEAVE A REPLY)**

**NEW QUESTION: 23**

Given the code fragment:

```
public static void main(String[] args) {
    String[] arr = {"A", "B", "C", "D"};
    for (int i = 0; i < arr.length; i++) {
        System.out.print(arr[i] + " ");
        if (arr[i].equals("C")) {
            continue;
        }
        System.out.println("Work done");
        break;
    }
}
```

What is the result?

**A.** A Work done

**B.** Compilation fails

**C.** A B C Work done

**D.** A B C D Work done

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 24**

Given the code snippet from a compiled Java source file:

```
public class MyFile
{
    public static void main (String[] args)
    {
        String arg1 = args[0];
        String arg2 = args[1];
        String arg3 = args[2];
        System.out.println("Arg is " + arg3);
    }
}
```

and this output:

```
Arg is 2
```

Which command should you run to obtain this output?

**A.** java MyFile 1 2 3 4
**B.** java MyFile
2
**C.** java MyFile 2 2
**D.** java MyFile 1 2 2
**Answer: (SHOW ANSWER)**

**NEW QUESTION: 25**
Given the code fragment from three files:

SalesMan.java:

```
package sales;
public class SalesMan { }
```

Product.java:

```
package sales.products;
public class Product { }
```

Market.java:

```
1. package market;
2. // insert code here
3. public class USMarket {
4.     SalesMan sm;
5.     Product p;
6. }
```

Which code fragment, when inserted at line 2, enables the code to compile?

A)
```
import sales.*;
```

B)
```
import java.sales.products.*;
```

C)
```
import sales;
import sales.products;
```

D)
```
import sales.*;
import products.*;
```

E)
```
import sales.*;
import sales.products.*;
```

**A.** Option D
**B.** Option C
**C.** Option A
**D.** Option E
**E.** Option B
**Answer: D (LEAVE A REPLY)**

**NEW QUESTION: 26**

Given the code from the Greeting.Java file:

```
public class Greeting {
    public static void main(String[] args) {
        System.out.println("Hello " + args[0]);
    }
}
```

Which set of commands prints Hello Duke in the console?

A)
```
javac Greeting
java Greeting Duke
```

B)
```
javac Greeting.java Duke
java Greeting
```

C)
```
javac Greeting.java
java Greeting Duke
```

D)
```
javac Greeting.java
java Greeting.class Duke
```

**A.** Option B

**B.** Option A

**C.** Option D

**D.** Option C

**Answer: D (LEAVE A REPLY)**

**NEW QUESTION: 27**

Given:

```
class Product {
    double price;
}

public class Test {
    public void updatePrice(Product product, double price) {
        price = price * 2;
        product.price = product.price + price;
    }
    public static void main(String[] args) {
        Product prt = new Product();
        prt.price = 200;
        double newPrice = 100;

        Test t = new Test();
        t.updatePrice(prt, newPrice);
        System.out.println(prt.price + " : " + newPrice);
    }
}
```

What is the result?

**A.** 400.0 : 100.0

**B.** 400.0 : 200.0
**C.** 200.0 : 100.0
**D.** Compilation fails.
**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 28**
Given the code fragment:

```
public static void main(String[] args) {
    LocalDate date = LocalDate.of(2012, 1, 30);
    date.plusDays(10);
    System.out.println(date);
}
```

What is the result?
**A.** 2012-02-10 00:00
**B.** 2012-01-30
**C.** 2012-02-10
**D.** A DateTimeExceptionis thrown at runtime.
**Answer: B (LEAVE A REPLY)**
Explanation/Reference:
Explanation:


**NEW QUESTION: 29**
Which three statements describe the object-oriented features of the Java language? (Choose three.)
**A.** Objects cannot be reused.
**B.** A subclass can inherit from a superclass.
**C.** Objects can share behaviors with other objects.
**D.** A package must contain more than one class.
**E.** Object is the root class of all other objects.
**F.** A main method must be declared in every class.
**Answer: (SHOW ANSWER)**
Explanation/Reference:

Reference: http://www.javaworld.com/article/2075459/java-platform/java-101--object-oriented-language- basics--part-5--object-and-its-methods.html (see the sub title, Object is root of all classes not all other objects)

**NEW QUESTION: 30**

Given the code fragment:

```
public static void main (String [] args) {
    String myStr = "Hello World";
    myStr.trim ()
    int i1 = myStr.indexOf (" ");
    System.out.printLn (i1);
}
```

What is the result?

**A.** An exception is thrown at runtime.

**B.** -1

**C.** 0

**D.** 5

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 31**

Given:

```
public class App {
    int count;
    public static void displayMsg () {
        count++;                                          // line n1
    System.out.println ("Welcome "+"Visit Count: "+count); // line n2
    }
    public static void main (String [] args) {
        App.displayMsg ();                                // line n3
        App.displayMsg ();                                // line n4
    }
}
```

What is the result?

**A.** Compilation fails at line n1 and line n2.

**B.** Welcome Visit Count:1
Welcome Visit Count: 1

**C.** Welcome Visit Count:1
Welcome Visit Count: 2

**D.** Compilation fails at line n3 and line n4.

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 32**
Given the following class declarations:
public abstract class Animal
public interface Hunter
public class Cat extends Animal implements Hunter
public class Tiger extends Cat
Which answer fails to compile?

A)
```
ArrayList<Animal> myList = new ArrayList<>();
myList.add(new Tiger());
```

B)
```
ArrayList<Hunter> myList = new ArrayList<>();
myList.add(new Cat());
```

C)
```
ArrayList<Hunter> myList = new ArrayList<>();
myList.add(new Tiger());
```

D)
```
ArrayList<Tiger> myList = new ArrayList<>();
myList.add(new Cat());
```

E)
```
ArrayList<Animal> myList = new ArrayLIst<>();
myList.add(new Cat());
```

**A.** Option B
**B.** Option A
**C.** Option E
**D.** Option D
**E.** Option C
**Answer: (SHOW ANSWER)**

**NEW QUESTION: 33**
Given:

```
public static void main(String[] args) {
    String ta = "A ";
    ta = ta.concat("B ");
    String tb = "C ";
    ta = ta.concat(tb);
    ta.replace('C', 'D');
    ta = ta.concat(tb);
    System.out.println(ta);
}
```

What is the result?

**A.** A B C D

**B.** A B D

**C.** A C D

**D.** A B D C

**E.** A B C C

**Answer: E (LEAVE A REPLY)**

**NEW QUESTION: 34**

Given:

```
public class Test {
    public static void main(String[] args) {
        boolean a = new Boolean(Boolean.valueOf (args[0]));
        boolean b = new Boolean(args[1]);
        System.out.println(a + " " + b);
    }
}
```

And given the commands:

```
javac Test.java
java Test TRUE null
```

What is the result?

**A.** true true

**B.** TRUE null

**C.** A ClassCastExceptionis thrown at runtime.

**D.** false false

**E.** true false

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 35**

Given:

```
public class TestScope {
    public static void main(String[] args) {
        int var1 = 200;
        System.out.print(doCalc(var1));
        System.out.print(" "+var1);
    }
    static int doCalc(int var1){
        var1 = var1 * 2;
        return var1;
    }
}
```

What is the result?

**A.** 400 200

**B.** 400 400

**C.** 200 200

**D.** Compilation fails.

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 36**

You are asked to develop a program for a shopping application, and you are given the following information:

The application must contain the classes Toy, EduToy, and ConsToy. The Toy class is the superclass

of the other two classes.

The int calculatePrice (Toy t) method calculates the price of a toy.

The void printToy (Toy t) method prints the details of a toy.

Which definition of the Toy class adds a valid layer of abstraction to the class hierarchy?

A:

```
public abstract class Toy{
    public abstract int calculatePrice(Toy t);
    public void printToy(Toy t) { /* code goes here */ }
}
```

B:

```
public abstract class Toy {
    public int calculatePrice(Toy t) ;
    public void printToy(Toy t) ;
}
```

C:

```
public abstract class Toy {
    public int calculatePrice(Toy t);
    public final void printToy(Toy t){ /* code goes here */ }
}
```

D:

```
public abstract class Toy {
    public abstract int calculatePrice(Toy t) { /* code goes here */ }
    public abstract void printToy(Toy t) { /* code goes here */ }
}
```

**A.** Option A
**B.** Option D
**C.** Option B
**D.** Option C
**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 37**
Given the code fragment:

```
abstract class Toy {
      int price;
      // line n1
}
```

Which three code fragments are valid at line n1? (Choose three.)
A:

```
public static void insertToy() {
      /* code goes here */
}
```

B:

```
public abstract Toy getToy() {
      return new Toy();
}
```

C:

```
public void printToy();
```

D:

```
public int calculatePrice() {
      return price;
}
```

E:

```
public abstract int computeDiscount();
```

**A.** Option B
**B.** Option A
**C.** Option C
**D.** Option D
**E.** Option E
**Answer: (SHOW ANSWER)**

**NEW QUESTION: 38**
Given the code fragment:

```
public static void main(String[] args) {
    double discount = 0;
    int qty = Integer.parseInt(args[0]);
    //line n1;
}
```

And given the requirements:

If the value of the qty variable is greater than or equal to 90, discount = 0.5 If the value of the qty variable is between 80 and 90, discount = 0.2 Which two code fragments can be independently placed at line n1 to meet the requirements? (Choose two.)

☐ A)
```
if (qty >= 90) { discount = 0.5; }
if (qty > 80 && qty < 90) { discount = 0.2; }
```

☐ B)
```
discount = (qty >= 90) ? 0.5 : 0;
discount = (qty > 80) ? 0.2 : 0;
```

☐ C)
```
discount = (qty >= 90) ? 0.5 : (qty > 80)? 0.2 : 0;
```

☐ D)
```
if (qty > 80 && qty < 90) {
    discount = 0.2;
} else {
    discount = 0;
}
if (qty >= 90) {
    discount = 0.5;
} else {
    discount = 0;
}
```

☐ E)
```
discount = (qty > 80) ? 0.2 : (qty >= 90) ? 0.5 : 0;
```

**A.** Option E

**B.** Option D

**C.** Option C

**D.** Option B

**E.** Option A

**Answer: C,E (LEAVE A REPLY)**

**NEW QUESTION: 39**

Given the code fragment:

```
LocalDate Time dt= LocalDateTime.of (2014, 7, 31, 1, 1);
dt.plusDays (30);
dt. plusMonths (1);
System.out.print (dt format (DateTimeFormatter. ISO_DATE) );
```

What is the result?

**A.** 2014-07-31

**B.** 07-31-2014

**C.** 2014-09-30

**D.** An exception is thrown at runtime.

**Answer: D (LEAVE A REPLY)**

**NEW QUESTION: 40**

Given:

Acc.java:

```
package p1;
public class Acc {
    int p;
    private int q;
    protected int r;
    public int s;
}
```

Test.java:

```
package p2;
import p1.Acc;
public class Test extends Acc {
    public static void main(String[] args) {
        Acc obj = new Test();
    }
}
```

Which statement is true?

**A.** Only s is accessible by obj.

**B.** p, r, and s are accessible by obj.

**C.** Both p and s are accessible by obj.

**D.** Both r and s are accessible by obj.

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 41**

Given the code fragment:

```
 4. public static void main(String[] args) {
 5.     boolean opt = true;
 6.     switch (opt) {
 7.         case true:
 8.             System.out.print("True");
 9.             break;
10.         default:
11.             System.out.print("***");
12.     }
13.     System.out.println("Done");
14. }
```

Which modification enables the code fragment to print TrueDone?

**A.** Replace line 5 With String opt = "true";

Replace line 7 with case "true":

**B.** At line 9, remove the break statement.

**C.** Remove the default section.

**D.** Replace line 5 with boolean opt = l;

Replace line 7 with case 1:

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 42**

Given the following main method:

```
public static void main(String[] args) {
    int num = 5;
    do {
        System.out.print(num-- +" ");
    } while(num == 0);
}
```



What is the result?

**A.** Nothing is printed

**B.** 5

**C.** 5 4 3 2 1

**D.** 5 4 3 2 1 0

**E.** 4 2 1

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 43**

Given the code fragment:

```
public static void main(String[] args) {
    List<String> names = new ArrayList<>();
    names.add("Robb");
    names.add("Bran");
    names.add("Rick");
    names.add("Bran");

    if (names.remove("Bran")) {
        names.remove("Jon");
    }
    System.out.println(names);
}
```



What is the result?

**A.** [Robb, Rick, Bran]

**B.** [Robb, Rick]

**C.** [Robb, Bran, Rick, Bran]

**D.** An exception is thrown at runtime.

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 44**

Given the code fragment:

```
public static void main(String[] args) {
    String date = LocalDate
                  .parse("2014-05-04")
                  .format(DateTimeFormatter.ISO_DATE_TIME);
    System.out.println(date);
}
```

What is the result?

**A.** May 04, 2014T00:00:00.000

**B.** 2014-05-04T00:00: 00. 000

**C.** An exception is thrown at runtime.

**D.** 5/4/14T00:00:00.000

**Answer: C (LEAVE A REPLY)**

**NEW QUESTION: 45**

Which is true about the switchstatement?

**A.** Its expression can evaluate to a collection of values.

**B.** The breakstatement, at the end of each case block, is optional.

**C.** Its caselabel literals can be changed at runtime.

**D.** It must contain the defaultsection.

**Answer: B (LEAVE A REPLY)**

Explanation/Reference:

Reference: https://www.geeksforgeeks.org/switch-statement-in-java/

**NEW QUESTION: 46**

Given:

```
class Equal {
      public static void main (String [] args) {
      String str1 = "Java";
      String [] str2 = { "J", "a", "v", "a"};
      String str3 = "";
       for (String str : str2) {
             str3 = str3+str;
       }
       boolean b1 = (str1= = str3);
       boolean b2 = (str1.equals (str3));
       System.out.print (b1+",   "+b2);
    }
}
```

What is the result?

**A.** false, false
**B.** true, true
**C.** false, true
**D.** true, false
**Answer: C (LEAVE A REPLY)**



**NEW QUESTION: 47**
Given the code fragment:

```
public static void main(String[] args) {
      LocalDate date = LocalDate.of(2012, 01, 32);
      date.plusDays(10);
      System.out.println(date);
}
```



What is the result?
**A.** Compilation fails
**B.** 2012-02-10
**C.** A DateTimeException is thrown at runtime.
**D.** 2012-02-11
**Answer: C (LEAVE A REPLY)**

**NEW QUESTION: 48**
Given the code fragment:

```
public class Employee {
    String name;
    boolean contract;
    double salary;
    Employee() {
        // line n1
    }
    public String toString(){
        return name + ":" + contract + ":" + salary;
    }
    public static void main(String[] args) {
        Employee e = new Employee();
        // line n2
        System.out.print(e);
    }
}
```

Which two modifications, when made independently, enable the code to print joe:true: 100.0? (Choose two.)

☐ A) Replace line n2 with:

```
e.name = "Joe";
e.contract = true;
e.salary = 100;
```

☐ B) Replace line n2 with:

```
this.name = "Joe";
this.contract = true;
this.salary = 100;
```

☐ C) Replace line n1 with:

```
this.name = new String("Joe");
this.contract = new Boolean(true);
this.salary = new Double(100);
```

☐ D) Replace line n1 with:

```
name = "Joe";
contract = TRUE;
salary = 100.0f;
```

☐ E) Replace line n1 with:

```
this("Joe", true, 100);
```

**A.** Option D
**B.** Option E
**C.** Option A
**D.** Option B
**E.** Option C
**Answer: (SHOW ANSWER)**

**NEW QUESTION: 49**
Given:

```
interface Readable {
    public void readBook();
    public void setBookMark();
}

abstract class Book implements Readable {    // line n1
    public void readBook() { }
    // line n2
}

class EBook extends Book {                   // line n3
    public void readBook() { }
    // line n4
}
```

And given the code fragment:

Book book1 = new EBook ();

Book1.readBook();

Which option enables the code to compile?

A) Replace the code fragment at `line n1` with:
`class Book implements Readable {`

B) At `line n2` insert:
`public abstract void setBookMark();`

C) Replace the code fragment at `line n3` with:
`abstract class EBook extends Book {`

D) At `line n4` insert:
`public void setBookMark() { }`

**A.** Option D

**B.** Option A

**C.** Option C

**D.** Option B

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 50**

Which three statements are true about the structure of a Java class? (Choose three.)

**A.** The methods are mandatory components of a class.

**B.** The fields need not be initialized before use.

**C.** A method can have the same name as a field.

**D.** A class can have only one private constructor.

**E.** A class can have overloaded static methods.

**F.** A public class must have a main method.

**Answer: A,C,F (LEAVE A REPLY)**

**NEW QUESTION: 51**

Given:

```
System.out.println("5 + 2 = " + 3 + 4);
System.out.println("5 + 2 = " + (3 + 4));
```

What is the result?

A)
```
5 + 2 = 34
5 + 2 = 34
```

B)
```
5 + 2 + 3 + 4
5 + 2 = 7
```

C)
```
7 = 7
7 + 7
```

D)
```
5 + 2 = 34
5 + 2 = 7
```

**A.** Option A

**B.** Option B

**C.** Option C

**D.** Option D

**Answer: D (LEAVE A REPLY)**

**NEW QUESTION: 52**

Given the code fragment:

```
public static void main(String[] args) {
    int[][] arr = new int[2][4];

    arr[0] = new int[]{1, 3, 5, 7};
    arr[1] = new int[]{1, 3};

    for (int[] a : arr) {
        for (int i=0; i < arr.length; i++) {
            System.out.print(a[i] + " ");
        }
        System.out.println();
    }
}
```

What is the result?

**A.** 1 3 5 7

1 3

**B.** 1 2

1 3

**C.** 1 3

1 3 0 0

**D.** 1 3

followed by an ArrayIndexOutOfBoundsException

**E.** Compilation fails.

**Answer: B (LEAVE A REPLY)**

Explanation/Reference:

Explanation:

```
1  class Main {
2    public static void main(String[] args) {
3      int[][] arr = new int[2][4];
4
5      arr[0] = new int[] {1, 2, 3, 5, 7};
6      arr[1] = new int[] {1, 3};
7
8      for (int[] a : arr) {
9        for (int i=0; i <arr.length; i++){
10         System.out.print (a[i] + " ");
11       }
12       System.out.println();
13     }
14     }
```

```
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
> javac -classpath .:/run_dir/junit-4.12.jar:/run_dir/hamcrest-c
ore-1.3.jar:/run_dir/json-simple-1.1.1.jar -d . Main.java
> java -classpath .:/run_dir/junit-4.12.jar:/run_dir/hamcrest-co
re-1.3.jar:/run_dir/json-simple-1.1.1.jar Main
1 2
1 3
```



**NEW QUESTION: 53**

Given:

```
package clothing;
public class Shirt {
    public statuc String getColor() {
        return "Green";
    }
}
```

Given the code fragment:

```
package clothing.pants;
// line n1
public class Jeans {
    public void matchShirt(){
        //line n2
        if(color.equals("Green")) {
            System.out.print("Fit")
        }
    }
    public static void main (String[] args) {
        Jeans trouser = new Jeans();
        trouser.matchShirt();
    }
}
```

Which two sets of actions, independently, enable the code fragment to print Fit?

**A.** At line n1 no changes required.

At line n2 insert: String color = Shirt.getColor();

**B.** At line n1 insert: import clothing;

At line n2 insert: String color = Shirt.getColor();

**C.** At line n1 insert: import clothing.Shirt;
At line n2 insert: String color = getColor();

**D.** At line n1 insert: import static clothing.Shirt.getcolor;
At line n2 insert: String color = getColor();

**E.** At line n1 insert: import clothing.*;
At line n2 insert: String color = Shirt.getColor();

**Answer: C (LEAVE A REPLY)**

**NEW QUESTION: 54**

Given the code fragment:

```
String[] strs = new String[2];
int idx = 0;
for (String s : strs) {
      strs[idx].concat(" element " + idx);
      idx++;
}
for (idx = 0; idx < strs.length; idx++) {
      System.out.println(strs[idx]);
}
```

What is the result?

**A.** Null
Null

**B.** A NullPointerException is thrown at runtime.

**C.** Null element 0
Null element 1

**D.** Element 0
Element 1

**Answer: B (LEAVE A REPLY)**

**NEW QUESTION: 55**

Given the code fragment:

```
public static void main (String[] args) {
      String[] arr = ("Hi", "How", "Are", "You");
      List<String> arrList = new ArrayList<>(Arrays.asList(arr);
      if (arrList.removeIf((String s) -> (return s.length() <= 2;))) {
            System.out.println(s + "removed")'
      }
}
```

What is the result?

**A.** The program compiles, but it prints nothing.

**B.** Hi removed

**C.** Compilation fails.

**D.** An UnsupportedOperationException is thrown at runtime.

**Answer: C (LEAVE A REPLY)**

**NEW QUESTION: 56**

Given the code fragment:

```
int num[][] = new int[1][3];
for (int i = 0; i < num.length; i++) {
    for (int j = 0; j < num[i].length; j++) {
        num[i][j] = 10;
    }
}
```

Which option represents the state of the num array after successful completion of the outer loop?

A:

```
num[0][0]=10
num[0][1]=10
num[0][2]=10
```

B:

```
num[0][0]=10
num[1][0]=10
num[2][0]=10
```

C:

```
num[0][0]=10
num[0][1]=0
num[0][2]=0
```

D:

```
num[0][0]=10
num[0][1]=10
num[0][2]=10
num[0][3]=10
num[1][0]=0
num[1][1]=0
num[1][2]=0
num[1][3]=0
```

**A.** Option A

**B.** Option C

**C.** Option B

**D.** Option D

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 57**

Given the code fragment:

```
3. public static void main(String[] args) {
4.       int x = 5;
5.       while (isAvailable(x)) {
6.           System.out.print(x);
7.
8.       }
9. }
10.
11. public static boolean isAvailable(int x) {
12.     return x-- > 0 ? true : false;
13. }
```

Which modification enables the code to print 54321?

**A.** Replace line 12 with return (x > 0) ? false: true;

**B.** Replace line 6 with --x; and, at line 7, insert System.out.print (x);

**C.** Replace line 6 with System.out. print (--x) ;

**D.** At line 7, insert x --;

**Answer: D (LEAVE A REPLY)**

**NEW QUESTION: 58**

Given:

```
public class MainTest {

    public static void main(String[] args) {
        System.out.println("String main " + args[0]);
    }
}
```

and commands:

```
javac MainTest.java
java MainTest "1 2 3"
```

What is the result?

**A.** String main 123

**B.** An exception is thrown at runtime

**C.** String main 1 2 3

**D.** String main 1

**Answer: D (LEAVE A REPLY)**

**NEW QUESTION: 59**

Given:

```
public class App {

    public static void main(String[] args) {
        Boolean[] bool = new Boolean[2];

        bool[0] = new Boolean(Boolean.parseBoolean("true"));
        bool[1] = new Boolean(null);

        System.out.println(bool[0] + " " + bool[1]);
    }
}
```

What is the result?

**A.** True false

**B.** True null

**C.** Compilation fails

**D.** A NullPointerException is thrown at runtime

**Answer: A (LEAVE A REPLY)**

**NEW QUESTION: 60**

Given the following array:

```
int[] intArr = {8, 16, 32, 64, 128};
```

Which two code fragments, independently, print each element in this array? (Choose two.)

A:

```
for (int i : intArr) {
    System.out.print(intArr[i] +" ");
}
```

B:

```
for (int i : intArr) {
    System.out.print(i +" ");
}
```

C:

```
for (int i=0 : intArr) {
    System.out.print(intArr[i] +" ");
    i++;
}
```

D:

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(i +" ");
}
```

E:

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(intArr[i] +" ");
}
```

F:

```
for (int i; i < intArr.length; i++) {
    System.out.print(intArr[i] +" ");
}
```

**A.** Option B
**B.** Option D
**C.** Option C
**D.** Option E
**E.** Option A
**F.** Option F
**Answer: A,D (LEAVE A REPLY)**

**NEW QUESTION: 61**
Given:

```
class C2 {
    public void displayC2() {
        System.out.print("C2");
    }
}
interface I {
    public void displayI();
}
class C1 extends C2 implements I {
    public void displayI() {
        System.out.print("C1");
    }
}
```

And given the code fragment:

```
C2 obj1 = new C1();
I obj2 = new C1();

C2 s = obj2;
I t = obj1;

t.displayI();
s.displayC2()
```

What is the result?
**A.** Compilation fails
**B.** C1C1
**C.** C2C2
**D.** C1C2

**Answer: (SHOW ANSWER)**



**NEW QUESTION: 62**

Given the code fragment:

```
public static void main(String[] args) {
      int[] arr = (1, 2, 3, 4);
      int i = 0;
      do {
            System.out.print(arr[i] + " ");
            i++;
      } while (i < arr.length - 1);
}
```

What is the result?

**A.** 1 2 3 4

**B.** 1 2 3 4

followed by an ArrayIndexOutOfBoundsException

**C.** 1 2 3

**D.** Compilation fails.

**Answer: B (LEAVE A REPLY)**

**NEW QUESTION: 63**

The following grid shows the state of a 2D array:

| | | |
|---|---|---|
| 0 | 0 | |
| | X | 0 |
| | X | X |

This grid is created with the following code:

```
char[][] grid = new char[3][3];
grid[1][1] = 'X';
grid[0][0] = '0';
grid[2][1] = 'X';
grid[0][1] = '0';
grid[2][2] = 'X';
grid[1][2] = '0';
//line n1
```

Which line of code, when inserted in place of //line n1, adds an X into the grid so that the grid contains three consecutive X's?

**A.** grid[1][2] = 'X';

**B.** grid[2][0] = 'X';

**C.** grid[3][1] = 'X';

**D.** grid[1][3] = 'X';

**E.** grid[0][2] = 'X';

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 64**

Given the code fragment:

```
public static void main(String[] args) {
    int ii = 0;
    int jj = 7;
    for (ii = 0; ii < jj - 1; ii = ii + 2) {
        System.out.print(ii + " ");
    }
}
```

What is the result?

**A.** 0 2 4

**B.** Compilation fails

**C.** 0 2 4 6

**D.** 2 4

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 65**

Given:

```
class Animal {
    String type = "Canine";
    int maxSpeed = 60;

    Animal () {}

    Animal (String type, int maxSpeed) {
       this.type = type;
       this.maxSpeed = maxSpeed;
    }
}

class WildAnimal extends Animal {
    String bounds;

    WildAnimal (String bounds) {
       //line n1
}
    WildAnimal (String type, int maxSpeed,String bounds) {
       //line n2
    }
}
```



And given the code fragment:

```
7. WildAnimal wolf = new WildAnimal ("Long");
8. WildAnimal tiger = new WildAnimal ("Feline", 80, "Short");
9. System.out.println (wolf.type + " " + wolf.maxSpeed + " " +
wolf.bounds);
10. Sytem.out.println (tiger.type + " " + tiger.maxSpeed + " " +
tiger.bounds;
```



Which two modifications enable the code to print the following output? (Choose two.) Canine 60 Long Feline 80 Short

**A.** Replace line n1 with:

```
this ("Canine", 60);
this.bounds = bounds
```



**B.** Replace line n2 with:

```
super (type, maxSpeed);
this (bounds);
```



**C.** . Replace line n1 with:

```
super ();
this.bounds = bounds;
```

**D.** Replace line n1 with:

```
this.bounds = bounds;
super ();
```

**E.** Replace line n2 with:

```
super (type, maxSpeed);
this.bounds = bounds;
```

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 66**

Given the code fragment:

```
public static void main (String [] args) {
    ArrayList<Integer> points = new ArrayList<> ();
    points.add (1);
    points.add (2);
    points.add (3);
    points.add (4);
    points.add (null);
    points.remove (2);
    points.remove (null);
    System.out.println(points);
}
```

What is the result?

**A.** A NullPointerException is thrown at runtime.

**B.** [1, 2, 4]

**C.** [1, 2, 4, null ]

**D.** [1, 3, 4, null ]

**E.** [1, 3, 4 ]

**F.** Compilation fails.

**Answer: (SHOW ANSWER)**

Explanation/Reference:

Explanation:


**NEW QUESTION: 67**

Given the following classes:

```
public class Employee {
    public int salary;
}

public class Manager extends Employee {
    public int budget;
}

public class Director extends Manager {
    public int stockOptions;
}
```

And given the following `main` method:

```
public static void main(String[] args) {
    Employee employee = new Employee();
    Manager  manager  = new Manager();
    Director director = new Director();
    //line n1
}
```



Which two options fail to compile when placed at line n1 of the main method? (Choose two.)

**A.** director.stockOptions = 1_000;

**B.** director.salary = 80_000;

**C.** manager.budget = 1_000_000;

**D.** employee.salary = 50_000;

**E.** employee.budget = 200_000;

**F.** manager.stockOption = 500;

**Answer: E,F (LEAVE A REPLY)**

**NEW QUESTION: 68**

Given:

```
class Test
      int a1;

      public static void doProduct(int a) {
            a = a * a;
      )

      public static void doString(StringBuilder s) {
            s.append(" " + s);
      }

      public static void main(String[] args) {
            Test item = new Test();
            item.a1 = 11;
            StringBuilder sb = new StringBuilder("Hello");
            Integer i = 10;
            doProduct(i);
            doString(sb);
            doProduct(item.a1);
            System.out.println(i + " " + sb + " " + item.a1);
      }
}
```



What is the result?

**A.** 100 Hello 121

**B.** 10 Hello Hello 121

**C.** 10 Hello 11

**D.** 10 Hello Hello 11

**E.** 100 Hello Hello 121

**Answer: B (LEAVE A REPLY)**

**NEW QUESTION: 69**

Given the code fragment:

```
public class Test {

    static int count = 0
    int i = 0;

    public void changeCount () {
        while (i<5) {
            i++;
            count++;
        }
    }

    public static void main (String [] args) {
        Test check1 = new Test ();
        Test check2 = new Test ();
        check1.changeCount ();
        check2.changeCount ();
        System.out. print (check1.count + " : " + check2.count);
    }
}
```



What is the result?

**A.** 5 : 5

**B.** 10 : 10

**C.** 5 : 10

**D.** Compilation fails.

**Answer: B (LEAVE A REPLY)**

Explanation/Reference:

Reference:

Version - JDK 1.8.0_66

Your Code ...

```
1  public class Test {
2
3          static int count = 0 ;
4          int i = 0;
5
6          public void changecount () {
7              while (i<5) {
8                  i++;
9                  count++;
10             }
11         }
12     public static void main (String [ ] args) {
13         Test check1 = new Test () ;
14         Test check2 = new Test () ;
15         check1.changecount () ;
16         check2.changecount () ;
17         System.out. print (check1.count + " : " + check2.count) ;
18     }
19 }
20
```

External Libraries ... Add External Library (from Maven Repo)

csl.keyboard

Input Arguments (args of Main Method)...

Interactive mode : OFF

Stdin Inputs...

Execute | Save | My Projects | Recent | Collaborate | Others | Goto Another Language/DB

Result...

*compiled and executed in 1.357 second(s)*

```
10 : 10
```



**NEW QUESTION: 70**

Given the code fragment:

```
3. public static void main(String[] args) {
4.     int iVar = 100;
5.     float fVar = 100.100f;
6.     double dVar = 123;
7.     iVar = fVar;
8.     fVar = iVar;
9.     dVar = fVar;
10.    fVar = dVar;
11.    dVar = iVar;
12.    iVar = dVar;
13. }
```



Which three lines fail to compile? (Choose three.)

**A.** Line 10
**B.** Line 11
**C.** Line 8
**D.** Line 7
**E.** Line 9
**F.** Line 12
**Answer: (SHOW ANSWER)**

**NEW QUESTION: 71**
Given:

```
class Vehicle {
    int x;
    Vehicle(){
        this(10);  // line n1
    }
    Vehicle(int x) {
        this.x = x;
    }
}

class Car extends Vehicle {
    int y;
    Car() {
        super();
        this(20);  // line n2
    }
    Car(int y) {
       this.y = y;
    }
    public String toString() {
        return super.x + ":" + this.y;
    }
}
```

And given the code fragment:

And given the code fragment:

```
Vehicle y = new Car();
System.out.println(y);
```

What is the result?
**A.** 10:20
**B.** Compilation fails at line n2
**C.** Compilation fails at line n1
**D.** 0:20
**Answer: B (LEAVE A REPLY)**

**NEW QUESTION: 72**

Given the code fragment:

```
13. List colors = new ArrayList();
14. colors.add("green");
15. colors.add("red");
16. colors.add("blue");
17. colors.add("yellow");
18. colors.remove(2);
19. colors.add(3, "cyan");
20. System.out.print(colors);
```

What is the result?

**A.** (green, blue, yellow, cyan)

**B.** (green, red, cyan, yellow)

**C.** (green, red, yellow, cyan)

**D.** An IndexOutOfBoundsExceptionis thrown at runtime.

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 73**

Which statement is true about the switch statement?

**A.** It must contain the default section.

**B.** The break statement, at the end of each case block, is mandatory.

**C.** Its case label literals can be changed at runtime.

**D.** Its expression must evaluate to a single value.

**Answer: (SHOW ANSWER)**

Explanation/Reference:

Reference: http://www.dummies.com/programming/java/switch-statements-in-java/

**NEW QUESTION: 74**

Given the following class:

```
public class Rectangle {
      private double length;
      private double height;
      private double area;

      public void setLength(double length) {
            this.length = length;
      }
      public void setHeight(double height) {
            this.height = height;
      }
      public void setArea() {
            area = length*height;
      }
}
```

Which two changes would encapsulate this class and ensure that the area field is always equal to length heightwhenever the Rectangle class is used?

**A.** Change the setArea method to private.

**B.** Call the setArea method at the beginning of the setLength method.

**C.** Change the area field to public.

**D.** Call the setArea method at the beginning of the setHeight method.

**E.** Call the setArea method at the end of the setLength method.

**F.** Call the setArea method at the end of the setHeight method.

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 75**

Given:

```
public class Test {
      public static void main(String[] args) {
            Test ts = new Test();
            System.out.print(isAvailable + " ");
            isAvailable= ts.doStuff();
            System.out.println(isAvailable);
      }
      public static boolean doStuff() {
            return !isAvailable;
      }
      static boolean isAvailable = false;
}
```

What is the result?

**A.** Compilation fails.

**B.** false false

**C.** true true

**D.** true false

**E.** false true

**Answer: E (LEAVE A REPLY)**

**NEW QUESTION: 76**

You are asked to create a method that accepts an array of integers and returns the highest value from that array.

Given the code fragment:

```
class Test {
    public static void main (String [] args) {
        int numbers [] = {12, 13, 42, 32, 15, 156, 23, 51, 12};
        int max = findMax (numbers);
}
/*line n1 */ {
   int max = 0;
   /* code goes here*/
   return max;
 }
}
```

Which method signature do you use at line n1?

**A.** final int findMax (int [] )

**B.** static int[] findMax (int max)

**C.** static int findMax (int [] numbers)

**D.** public int findMax (int [] numbers)

**Answer: (SHOW ANSWER)**



**NEW QUESTION: 77**

Given the code fragment:

```
public static void main(String[] args) {
    Short s1 = 200;
    Integer s2 = 400;
    Long s3 = (long) s1 + s2;          //line n1
    String s4 = (String) (s3 * s2);    //line n2
    System.out.println("Sum is " + s4);
}
```

What is the result?

**A.** A ClassCastException is thrown at line n1.

**B.** Compilation fails at line n1.

**C.** Sum is 600

**D.** Compilation fails at line n2.

**E.** A ClassCastException is thrown at line n2.

**Answer: D (LEAVE A REPLY)**

**NEW QUESTION: 78**

Given:

```
class Vehicle {
    String type = "4W";
    int maxSpeed = 100;

    Vehicle(String type, int maxSpeed) {
        this.type = type;
        this.maxSpeed = maxSpeed;
    }
}

class Car extends Vehicle {
    String trans;

    Car(String trans) {          //line n1
        this.trans = trans;
    }

    Car(String type, int maxSpeed, String trans) {
        super(type, maxSpeed);
        this(trans);             //line n2
    }
}
```

And given the code fragment:

```
 7. Car c1 = new Car("Auto");
 8. Car c2 = new Car("4W", 150, "Manual");
 9. System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);
10. System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);
```

What is the result?

**A.** Compilation fails at both line n1 and line n2

**B.** Compilation fails only at line n2

**C.** 4W 100 Auto
4W 150 Manual

**D.** Compilation fails only at line n1

**E.** Null 0 Auto
4W 150 Manual

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 79**

Given the code fragment:

```
int[] array = {1, 2, 3, 4, 5};
```

And given the requirements:

1. Process all the elements of the array in the order of entry.

2. Process all the elements of the array in the reverse order of entry.
3. Process alternating elements of the array in the order of entry.
Which two statements are true? (Choose two.)
**A.** Requirement 3 CANNOT be implemented by using either the enhanced for loop or the standard for loop.
**B.** Requirement 1 can be implemented by using the enhanced for loop.
**C.** Requirements 2 and 3 CANNOT be implemented by using the standard for loop.
**D.** Requirements 1, 2, and 3 can be implemented by using the enhanced for loop.
**E.** Requirements 1, 2, and 3 can be implemented by using the standard for loop.
**Answer: (SHOW ANSWER)**

**NEW QUESTION: 80**
Given the code fragment:

```
String[] arr = {"Hi", "How", "Are", "You"};
List<String> arrList = new ArrayList<>(Arrays.asList(arr));
if(arrList.removeIf(s -> { System.out.print(s); return s.length()<=2;} )){
System.out.println(" removed");
}
```

What is the result?
**A.** Compilation fails.
**B.** The program compiles, but it prints nothing.
**C.** HiHowAreYou removed
**D.** An UnsupportedOperationExceptionis thrown at runtime.
**Answer: B (LEAVE A REPLY)**
Explanation/Reference:
Explanation:


**NEW QUESTION: 81**
Given:

```
class X {
    static int i;
    int j;
    public static void main(String[] args) {
        X x1 = new X();
        X x2 = new X();
        x1.i = 3;
        x1.j = 4;
        x2.i = 5;
        x2.j = 6;
        System.out.println(
            x1.i + " " +
            x1.j + " " +
            x2.i + " " +
            x2.j);
    }
}
```

What is the result?

**A.** 3 4 3 6

**B.** 5 4 5 6

**C.** 3 4 5 6

**D.** 3 6 4 6

**Answer: B (LEAVE A REPLY)**

**NEW QUESTION: 82**

Given:

```
public class Vowel {
    private char var;
    public static void main(String[] args) {
        char var1 = 'a';
        char var2 = var1;
        var2 = 'e';

        Vowel obj1 = new Vowel ();
        Vowel obj2 = obj1;
        obj1.var = 'i';
        obj2.var = 'o';

        System.out.println(var1 + ", " +var2);
        System.out.print(obj1.var + ", " +obj2.var);
    }
}
```

**A.** e, e

i, o

**B.** a, e

o, o

**C.** a, e
i, o
**D.** e, e
o, o
**Answer: B (LEAVE A REPLY)**

**NEW QUESTION: 83**
Given:

```
public class SumTest {

    public static void doSum(Integer x, Integer y) {
        System.out.println("Integer sum is " + (x + y));
    }

    public static void doSum(double x, double y) {
        System.out.println("double sum is " + (x + y));
    }

    public static void doSum(float x, float y) {
        System.out.println("float sum is " + (x + y));
    }

    public static void doSum(int x, int y) {
        System.out.println("int sum is " + (x + y));
    }

    public static void main(String[] args) {
        doSum(10, 20);
        doSum(10.0, 20.0);
    }
}
```

What is the result?
A:

```
int sum is 30
float sum is 30.0
```

B:

```
int sum is 30
double sum is 30.0
```

C:

```
integer sum is 30
double sum is 30.0
```

D:

```
integer sum is 30
float sum is 30.0
```

**A.** Option C
**B.** Option D

**C.** Option B
**D.** Option A
**Answer: B (LEAVE A REPLY)**

**NEW QUESTION: 84**
Given the code fragment:

```
public class Person {
    String name;
    int age = 25;

    public Person (String name) {
        this (); //                                  //line n1
        setName(name);
    }
    public Person (String name, int age) {
        Person (name);                              //line n2
        setAge (age);
    }
    //setter and getter methods go here

    public String show () {
        return name + " " + age;
    }
    public static void main (String [] args) {
        Person p1 = new Person ("Jesse");
        Person p2 = new Person ("Walter", 52);
        System.out.println (p1.show () );
        System.out.println (p2.show () );
    }
}
```



What is the result?
**A.** Jesse 25
Walter 52
**B.** Compilation fails at both line n1 and line n2.
**C.** Compilation fails only at line n1.
**D.** Compilation fails only at line n2.
**Answer: (SHOW ANSWER)**

**NEW QUESTION: 85**

What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

**A.** Encapsulation

**B.** Inheritance

**C.** Abstraction

**D.** Instantiation

**E.** Polymorphism

**Answer: A (LEAVE A REPLY)**

Explanation/Reference:

Explanation:

Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

Reference: http://www.tutorialspoint.com/java/java_access_modifiers.htm

**NEW QUESTION: 86**

Given the code fragment:

```
public static void main(String[] args) {
    String str = " ";
    str.trim();
    System.out.println(str.equals("") + " " + str.isEmpty());
}
```



What is the result?

**A.** true true

**B.** false false

**C.** false true

**D.** true false

**Answer: (SHOW ANSWER)**

**NEW QUESTION: 87**

Given the code fragment:

```
public static void main(String[] args) {
    String[][] arr = {{"A", "B", "C"}, {"D", "E"}};
    for (int i = 0; i < arr.length; i++) {
        for (int j = 0; j < arr[i].length; j++) {
            System.out.print(arr[i][j] + " ");
            if (arr[i][j].equals("B")) {
                break;
            }
        }
        continue;
    }
}
```



What is the result?

**A.** Compilation fails.
**B.** A B D E
**C.** A B C D E
**D.** A B C
**Answer: (SHOW ANSWER)**

**NEW QUESTION: 88**
You are developing a banking module. You have developed a class named ccMask that has a maskcc method.
Given the code fragment:

```
class CCMask {
    public static String maskCC(String creditCard) {
        String x = "XXXX-XXXX-XXXX-";
        //line n1
    }

    public static void main(String[] args) {
        System.out.println(maskCC("1234-5678-9101-1121"));
    }
}
```



You must ensure that the maskcc method returns a string that hides all digits of the credit card number except the four last digits (and the hyphens that separate each group of four digits).
Which two code fragments should you use at line n1, independently, to achieve this requirement? (Choose two.)

☐ A)
```
StringBuilder sb = new StringBuilder(creditCard);
sb.substring(15, 19);
return x + sb;
```

☐ B)
```
return x + creditCard.substring(15, 19);
```

☐ C)
```
StringBuilder sb = new StringBuilder(x);
sb.append(creditCard, 15, 19);
return sb.toString();
```

☐ D)
```
StringBuilder sb = new StringBuilder(creditCard);
StringBuilder s = sb.insert(0, x);
return s.toString();
```



**A.** Option A
**B.** Option C
**C.** Option B
**D.** Option D
**Answer: (SHOW ANSWER)**

**NEW QUESTION: 89**
Given:

```
class A {
   public void test () {
       System.out.println ("A");
   }
}
class B extends A {
    public void test () {
       System.out.println ("B");
   }
}
public class C extends A {
    public void test () {
       System.out.println ("C");
    }

    public static void main (String [] args) {
        A b1 = new A ();
        A b2 = new C ();

        b1 = (A) b2;                //line n1
        A b3 = (B) b2;              //line n2
        b1.test ();
        b3.test ();
    }
}
```



What is the result?

**A.** A ClassCastException is thrown only at line n1.

**B.** A ClassCastException is thrown only at line n2.

**C.** C
C

**D.** A
C

**E.** A
B

**Answer: D (LEAVE A REPLY)**