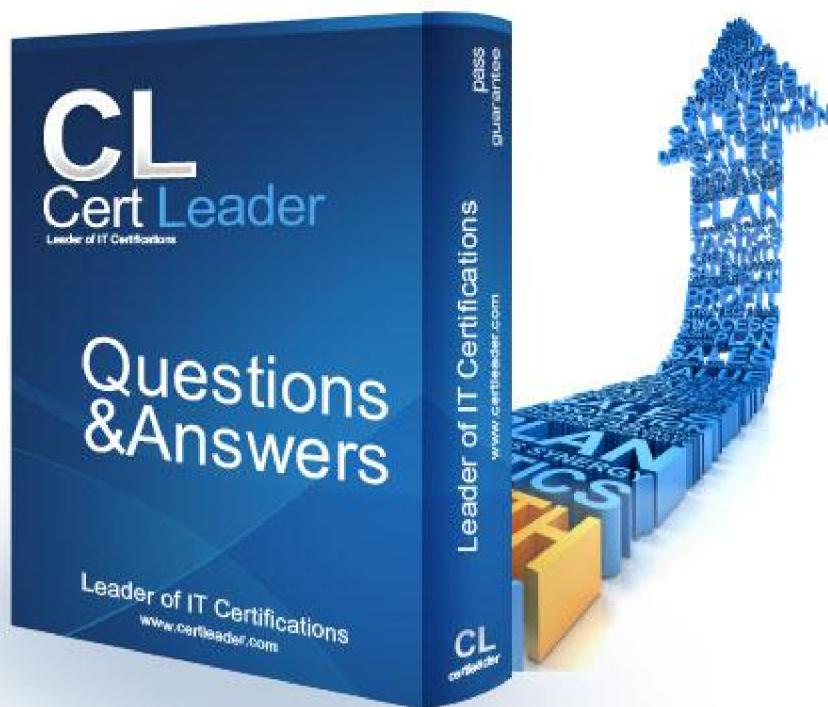


1Z0-808:

Java SE 8 Programmer I

Version:

V18.0



1. Given:

```
public class Test {  
    public static void main(String[] args) {  
        int x = 1;  
        int y = 1;  
        if(x++ < ++y){  
            System.out.print("Hello ");  
        } else {  
            System.out.print("Welcome ");  
        }  
        System.out.print("Log " + x + ":" + y);  
    }  
}
```

What is the result?

A. Hello

Log 2:2

B. Welcome

Log 1:2

C. Welcome

Log 2:1

D. Hello

Log 1:2

Answer: A

Explanation:

```
2
3 public class Test {
4
5     public static void main(String[] args) {
6
7         int x = 1;
8         int y = 1;
9
10        if (x++ < ++y) { // x is still 1, y is 2 ==> true
11            System.out.println("Hello ");
12        }else {
13            System.out.println("Welcome ");
14        }
15        System.out.println("Log " + x + ":" + y); // x is used again so x becomes 2
16
17    }
18
19 }
```

Hello
Log 2:2

2. Given:

```
public class App {  
    public static void main(String[] args) {  
        int i = 10;  
        int j = 20;  
        int k =(j += i)/ 5;  
        System.out.print(i + " : " + j + " : " + k);  
    }  
}
```

What is the result?

- A. 10 : 30 : 6
- B. 10 : 22 : 22
- C. 10 : 22 : 20
- D. 10 : 22 : 6

Answer: A

Explanation:

```
5 public static void main(String[] args) {  
6     int i = 10;  
7     int j = 20;  
8     int k = (j += i) / 5;    // k = 30/5 = 6  
9     // (j = j+i) = (j = 30) in parentheses j is equalized to 30  
10    System.out.println(i + " : " + j + " : " + k);  
11}
```

Console
<terminated> App (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Content
10 : 30 : 6

Other possible version:

```
4 public static void main(String[] args) {  
5     int i = 10;  
6     int j = 20;  
7     int k = j += i / 5 ;  
8     // k = j = j + i / 5 ==> first i / 5 = 2, then k = j = j + 2 => 22  
9     System.out.println(i + " : " + j + " : " + k);  
10    }  
11}
```

Console
<terminated> App_2 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/ja
10 : 22 : 22

3. 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. 5/4/14T00:00:00.000
- D. An exception is thrown at runtime.

Answer: D

Explanation:

```
8 public static void main(String[] args) {
9     String date = LocalDate
10        .parse("2014-05-04")
11        .format(DateTimeFormatter.ISO_DATE_TIME);
12     System.out.println(date);
13 }
14
15
```

Console > <terminated> Q003 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 20, 2020, 4:32:03 PM - 4:32:04 PM)

Exception in thread "main" java.time.temporal.UnsupportedTemporalTypeException: Unsupported field: HourOfDay

```
8 public static void main(String[] args) {
9     String date = LocalDate
10        .parse("2014-05-04") // time is not assigned here
11        .format(DateTimeFormatter.ISO_DATE); // _TIME should be removed
12     System.out.println(date);
13 }
14
15
```

Console > <terminated> Q003 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 20, 2020, 4:33:53 PM - 4:33:54 PM)

2014-05-04

4. Given the code fragment:

```
public static void main(String[] args) {  
    int[] stack = {10, 20, 30};  
    int size = 3;  
    int idx = 0;  
    /* line n1 */  
    System.out.print("The Top element: " + stack[idx]);  
}
```

Which code fragment, inserted at line n1, prints The Top element: 30?

A

```
do {  
    idx++;  
} while (idx >= size);
```

B

```
while (idx < size) {  
    idx++;  
}
```

C

```
do {  
    idx++;  
} while (idx < size - 1);
```

D

```
do {  
    idx++;  
} while (idx <= size);
```

E

```
while (idx <= size - 1) {  
    idx++  
}
```

A. Option A

B. Option B

C. Option C

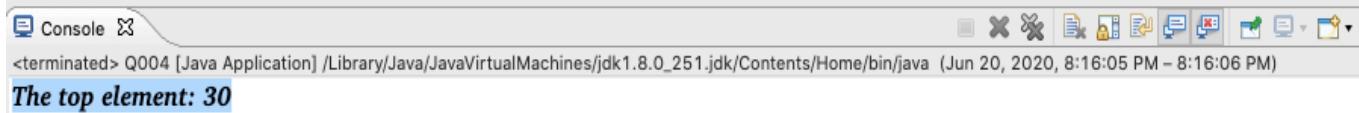
D. Option D

E. Option E

Answer: C

Explanation:

```
5④ public static void main(String[] args) {
6    int[] stack = {10, 20, 30};
7    int size =3;
8    int idx = 0;
9    do {           // Loop steps:
10       idx++;      // 1: idx = 0,  2: idx =1 ,
11    }while(idx < size-1); // 1: idx = 1,  2: idx =2, (2<2) is false loop ends --> The top element: 30
12    System.out.println("The top element: " + stack[idx]);
13 }
14 }
15 /*
16 * A.   do {           // Loop steps:
17     idx++;          // 1: idx = 0,
18 }while(idx >= size); // 1: idx = 1, (1>=3) is false loop ends --> The top element: 20
20
21 B.           // Loop steps:
22 while(idx < size){ // 1: idx = 0,  2: idx =1 ,  3: idx =2, 4: idx = 3 (3<3) is false loop ends
23     idx++;         // 1: idx =1,  2: idx =2,  3: idx =3,      --> ArrayIndexOutOfBoundsException
24 }
25
26 C.   do {           // Loop steps:
27     idx++;          // 1: idx = 0,  2: idx =1 ,
28 }while(idx < size-1); // 1: idx = 1,  2: idx =2, (2<2) is false loop ends --> The top element: 30
29
30 D.   do {           // Loop steps:
31     idx++;          // 1: idx = 0,  2: idx =1 ,  3: idx = 2  4: idx =3
32 }while(idx <= size); // 1: idx =1,  2: idx =2,  3: idx =3  4: idx = 4 (4<=3) is false loop ends
33                                     --> ArrayIndexOutOfBoundsException
34
35 E.           // Loop steps:
36 while(idx <= size -1){ // 1: idx = 0,  2: idx =1 ,  3: idx = 2  4: idx =3 (3<=2) is false loop ends
37     idx++;            // 1: idx =1,  2: idx =2,  3: idx =3      --> ArrayIndexOutOfBoundsException
38 */
39
40
```



The screenshot shows a Java application running in an IDE. The console window displays the output of the program: "The top element: 30". The IDE interface includes various toolbars and a status bar at the bottom.

```
Console <terminated> Q004 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 20, 2020, 8:16:05 PM – 8:16:06 PM)
The top element: 30
```

5. 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 optional.
 - C. Its case label literals can be changed at runtime.
 - D. Its expression must evaluate to a collection of values.

Answer: B

Explanation:

6. Given the code fragment:

```
4. class X {  
5.     public void printFileContent() {  
6.         /* code goes here */  
7.         throw new IOException();  
8.     }  
9. }  
10. public class Test {  
11.     public static void main(String[] args) {  
12.         X xobj = new X();  
13.         xobj.printFileContent();  
14.     }  
15. }
```

Which two modifications should you make so that the code compiles successfully? (Choose two.)

- A Replace line 13 with:

```
try {  
    xobj.printFileContent();  
}  
catch(Exception e) {}  
catch(IOException e) {}
```

- B Replace line 7 with throw IOException ("Exception raised");

- C Replace line 11 with public static void main(String[]) args) throws Exception {

- D At line 14, insert throw new IOException();

- E Replace line 5 with public void printFileContent() throws IOException {

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

Answer: C E

Explanation:

```
4 class X{  
5     public void printFileContent() throws IOException {  
6         /*code goes here */  
7         throw new IOException();  
8     }  
9 }  
10 public class Test {  
11     public static void main(String[] args) throws IOException {  
12         X xobj = new X();  
13         xobj.printFileContent();  
14     }  
15 }  
16 }
```

7. 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 5 6
- B.3 4 3 6
- C.5 4 5 6
- D.3 6 4 6

Answer: C

Explanation:

```
3 public class X {  
4     static int i; // static fields gets changed for every object  
5     int j;  
6     public static void main(String[] args) {  
7         X x1 = new X();  
8         X x2 = new X();  
9         x1.i= 3;  
10        x1.j = 4;  
11        x2.i= 5;  
12        x2.j = 6;  
13        System.out.println(  
14            x1.i + " " +  
15            x1.j + " " +  
16            x2.i + " " +  
17            x2.j);  
18    }  
19 }
```

Console >
<terminated> X (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Con
5 4 5 6

8. 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")) {  
                continue;  
            }  
            continue;  
        }  
    }  
}
```

What is the result?

- A. A B C
- B. A B C D E
- C. A B D E
- D. Compilation fails.

Answer: D

Explanation:

```
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.println(arr[i][j] + " ");  
            continue;  
        }  
    }  
    continue; //
```

}  continue cannot be used outside of a loop

Also **break;** cannot be used outside of a loop

Another possible version:

```
4  
5 public static void main(String[] args) {  
6     String [][] arr = {"A", "B", "C"}, {"D", "E"};  
7     for (int i = 0; i < arr.length; i++) {  
8         for (int j = 0; j < arr[i].length; j++) {  
9             System.out.print(arr[i][j] + " ");  
10            continue;  
11        }  
12    }  
13 }  
14 }
```

Console 

<terminated> Q008_2 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.

A B C D E

9. 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 A

B. Option B

C. Option C

D. Option D

Answer: C

Explanation:

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

public class DVD extends CD{
    int c;
    DVD(int r, int c){
        // line n1
        super(r);
        this.c = c;
    }

    public static void main(String[] args) {
        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;      // super.r is not correct way to call parent class's constructor
    this.c = c;

B. super(r);
    this(c);      // this(); should be the first statement right after constructor

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

D. this.c = r;
    super(c);      // super(); should be the first statement right after constructor
*/
```

10. 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. 200 200
- C. 400 400
- D. Compilation fails.

Answer: A

Explanation:

```
3 public class TestScope {  
4     public static void main(String[] args) {  
5         int var1 = 200;  
6         System.out.print(doCalc(var1)); // prints the returned var from doCalc method  
7         System.out.print(" " + var1); // prints the local var  
8     }  
9     static int doCalc(int var1) {  
10         var1 = var1 * 2;  
11         return var1;  
12     }  
13 }
```

Console >
<terminated> TestScope [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 21)
400 200

11. Given the code fragment:

```
String[] strs = {"A", "B"};
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. A
- B
- B. A element 0
- C. A NullPointerException is thrown at runtime.
- D. A 0
- B 1

Answer: A

Explanation:

```
5⊕ public static void main(String[] args) {  
6   String[] strs = {"A", "B"};  
7   int idx = 0;  
8   for (String s : strs) {  
9     strs[idx].concat(" element " + idx);  
10    idx++;  
11  }  
12  for (idx = 0; idx < strs.length; idx++) {  
13    System.out.println(strs[idx]);  
14  }  
15 }
```

Console
<terminated> Q011 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_261.jdk/Contents/Home/bin/java
A
B

12. 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 A

B. Option B

C. Option C

D. Option D

Answer: C

Explanation:

```
3 public class SumTest {  
4     public static void doSum(Integer x, Integer y) {  
5         System.out.println("Integer sum is " + (x + y));  
6     }  
7  
8     public static void doSum(double x, double y) {  
9         System.out.println("double sum is " + (x + y));  
10    }  
11  
12    public static void doSum(float x, float y) {  
13        System.out.println("float sum is " + (x + y));  
14    }  
15  
16    public static void doSum(int x, int y) {  
17        System.out.println("int sum is " + (x + y));  
18    }  
19  
20    public static void main(String[] args) {  
21        doSum(10, 20);;  
22        doSum(10.0, 20.0);  
23        // int --> float --> double --> Integer  
24        // if it was written as doSum(10.0f, 20.0f); it would go to float  
25    }
```

Console 

```
<terminated> SumTest (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Ho  
int sum is 30  
double sum is 30.0
```

13. 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 n2.
- B. Compilation fails at line n1.
- C. 20:20
- D. 10:20

Answer: C

Explanation:

```
3 class Vehicle{  
4     int x;  
5     Vehicle(){  
6         this(10);    // line n1  
7     }  
8     Vehicle(int x){  
9         this.x= x;  
10    }  
11 }  
12  
13 public class Car extends Vehicle{  
14     int y;  
15     Car(){  
16         super(10);    // line n2  
17     }  
18     Car(int y){  
19         super(y);  
20         this.y =y;  
21     }  
22     public String toString() {  
23         return super.x + ":" + this.y;  
24     }  
25     public static void main(String[] args) {  
26         Vehicle y = new Car(20);  
27         System.out.println(y);  
28     }  
29 }  
30 }
```

Console

<terminated> Car [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jc

20:20

14. Given the code fragment:

```
String str = "Sweet Sweat";
String str2 = str.trim().charAt(6) + "" +str.indexOf("Sw",1);
System.out.println(str2);
```

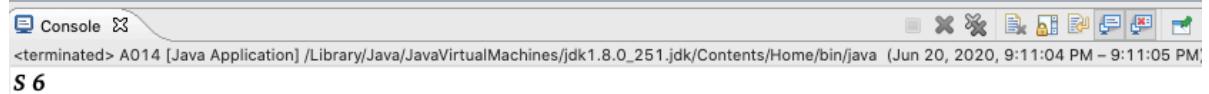
What is the result?

- A. S 6
- B. S 5
- C. s-1
- D. w 7

Answer: A

Explanation:

```
7
8 public static void main(String[] args) {
9     String str = "Sweet Sweat";
10    String str2 = str.trim().charAt(6) + " " + str.indexOf("Sw", 1); //
11    System.out.println(str2);
12 }
13 // no white space before and after "Sweat Sweat", so trim() is not effective
14 // 6th character in str is still S.      --  charAt(6) gives S
15 // str.indexOf("Sw", 1); means return the index number of characters "Sw" but start to count from 1|
```



15. Given the code fragment:

```
7.  StringBuilder sb1 = new StringBuilder("Duke");  
8.  String str1 = sb1.toString();  
9.  // insert code here  
10. System.out.print(str1 == str2);
```

Which code fragment, when inserted at line 9, enables the code to print true?

- A. String str2 = str1;
- B. String str2 = new String(str1);
- C. String str2 = sb1. toString();
- D. String str2 = "Duke";

Answer: A

Explanation:

```
7  StringBuilder sb1 = new StringBuilder("Duke");
8  String str1 = sb1.toString();
9  /* insert code here */ String str2 = str1; // the objects are equalized
10 System.out.println(str1 == str2); // == checks if the objects are equal, not their values
11 }
12 }
```

The screenshot shows a Java application window. The code editor contains the provided Java code. The console tab is active, displaying the output of the program. The output shows the word "true", indicating that the comparison between the two string objects resulted in true.

```
Console ✘
<terminated> Q015 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 20, 2020, 9:21:1
true
```

16. 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. An exception is thrown at runtime.
- B. 07-31-2014
- C. 2014-07-31
- D. 2014-09-30

Answer: A

Explanation:

Note: typos have been corrected!

```
8 public static void main(String[] args) {
9     LocalDateTime dt = LocalDateTime.of(2014, 7, 31, 1, 1);
10    dt = dt.plusDays(30);
11    dt = dt.plusMonths(1);
12    System.out.println(dt.format(DateTimeFormatter.ISO_DATE));
13 }
14
15 }
16 //What is the result?
17 //A. An exception is thrown at runtime. // if it was written with .parse()
18 //B. 07-31-2014 // LocalDateTime.of(2014, 7, 31, 1, 1); -- gives the format
19 //C. 2014-07-31
20 //D. 2014-09-30 // dt was not assigned as dt = dt.plus... --> false
21
```

Console



<terminated> Q016 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 20, 2021)

2014-09-30

17. 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 = 3, y = 5;  
        Test obj = new Test(x, y);  
        System.out.println(x + " " + y);  
    }  
}
```

What is the result?

- A. Compilation fails.
- B. 3 5
- C. 0 0
- D. 9 25

Answer: B

Explanation:

```
3 | public class Test {  
4 |     int x, y;  
5 |     public Test(int x, int y) {  
6 |         initialize(x, y);  
7 |     }  
8 |     public void initialize(int x, int y) {  
9 |         this.x = x * x;  
10 |        this.y = y * y;  
11 |    }  
12 |    public static void main(String[] args) {  
13 |        int x = 3, y = 5;  
14 |        Test objTest = new Test(x, y);  
15 |        System.out.println(x + " " + y); // prints the local variables  
16 |    }  
17 }
```

Console 
<terminated> Test (5) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/jav
3 5

18. Given:

```
class Vehicle {  
    String type = "4W";  
    int maxSpeed = 100;  
  
    Vehicle(String type, int maxSpeed) {  
        this.type = type;  
        this.maxSpeed = maxSpeed;  
    }  
    Vehicle() {}  
}  
  
class Car extends Vehicle {  
    String trans;  
  
    Car(String trans) {           //line n1  
        this.trans = trans;  
    }  
  
    Car(String type, int maxSpeed, String trans) {  
        super(type, maxSpeed);   // line n2  
        this.trans = trans;  
    }  
}
```

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. 4W 100 Auto

4W 150 Manual

B. null 0 Auto

4W 150 Manual

C. Compilation fails only at line n1

D. Compilation fails only at line n2

E. Compilation fails at both line n1 and line n2

Answer: A

Explanation:

```
3 class Vehicle{
4     String type = "4W";
5     int maxSpeed = 100;
6     Vehicle(String type, int maxSpeed){
7         this.type = type;
8         this.maxSpeed = maxSpeed;
9     }
10    Vehicle(){}
11 }
12
13 public class Car extends Vehicle {
14     String trans;
15
16     Car(String trans){      // line n1
17         this.trans = trans;
18     }
19     Car(String type, int maxSpeed, String trans){
20         super(type, maxSpeed); // line n2
21         this.trans = trans;
22     }
23     public static void main(String[] args) {
24         Car c1 = new Car("Auto");
25         Car c2 = new Car("4W", 150, "Manual");
26         System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);
27         System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);
28     }
29 }
```

Console ☰

<terminated> Car (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/jav

4W 100 Auto

4W 150 Manual

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

- A. A class cannot have the same name as its field.
- B. A public class must have a main method.
- C. A class can have final static methods.
- D. A class can have overloaded private constructors.
- E. Fields need to be initialized before use.
- F. Methods and fields are optional components of a class.

Answer: C D F

Explanation:

20. Given:

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

What is the result?

- A. 10 Hello Hello 11
- B. 10 Hello Hello 121
- C. 100 Hello 121
- D. 100 Hello Hello 121
- E. 10 Hello 11

Answer: E

Explanation:

```
3 public class Test {  
4     int a1;  
5     public static void doProduct(int a) {  
6         a = a * a;  
7     }  
8  
9     public static void doString(String s) {  
10        s.concat(" " + s);  
11    }  
12    public static void main(String[] args) {  
13        Test item = new Test();  
14        item.a1 = 11;  
15        String sb = "Hello";  
16        Integer i = 10;  
17        doProduct(i);  
18        doString(sb);  
19        doProduct(item.a1);  
20        System.out.println(i + " " + sb + " " + item.a1);  
21    }  
22}
```

Console
<terminated> Test (6) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/
10 Hello 11

21. Given:

```
class Vehicle {  
    Vehicle() {  
        System.out.println("Vehicle");  
    }  
}  
class Bus extends Vehicle {  
    Bus() {  
        System.out.println("Bus");  
    }  
}  
public class Transport {  
    public static void main (String[] args) {  
        Vehicle v = new Bus();  
    }  
}
```

What is the result?

A. Vehicle

 Bus

B. Bus

 Vehicle

C. Bus

D. The program doesn't print anything

Answer: A

Explanation:

```
2 class Vehicle {  
3     Vehicle(){  
4         System.out.println("Vehicle");  
5     }  
6 }  
7 class Bus extends Vehicle{  
8     Bus(){  
9         System.out.println("Bus");  
10    }  
11 }  
12 public class Transport {  
13     public static void main(String[] args) {  
14         Vehicle v = new Bus();  
15     }  
16 }
```

Console
<terminated> Transport [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8
Vehicle
Bus

22. Given the code fragment:

```
public static void main(String[] args) {  
    int ans;  
    try {  
        int num = 10;  
        int div = 0;  
        ans = num / div;  
    } catch (ArithmecticException ae) {  
        ans = 0; // line n1  
    } catch (Exception e) {  
        System.out.println("Invalid calculation");  
    }  
    System.out.println("Answer = " + ans); // line n2  
}
```

What is the result?

- A. Answer = 0
- B. Invalid calculation
- C. Compilation fails only at line n1.
- D. Compilation fails only at line n2.
- E. Compilation fails at line n1 and line2.

Answer: D

Explanation:

```
5 public static void main(String[] args) {
6     int ans; // if it is initialized, for example int ans = 0; --> Answer 0
7     try {
8         int num = 10;
9         int div = 0;
10        ans = num / div;
11    } catch (ArithmeticException ae) {
12        ans = 0; // line n1
13    } catch (Exception e) {
14        System.out.println("Invalid calculation");
15    }
16    System.out.println("Answer = " + ans); // line n2
17    // ans in the try-catch block can not be seen outside the block
18 }
19 }
```

Console <terminated> Q022 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 20, 2020)

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

The local variable ans may not have been initialized

at q022.Q022.main(Q022.java:16)

23. Given the code fragment:

```
6. char colorCode = 'y';
7. switch (colorCode) {
8.     case 'r':
9.         int color = 100;
10.    break;
11.   case 'b':
12.       color = 10;
13.       break;
14.   case 'y':
15.       color = 1;
16.       break;
17. }
18. System.out.println(color);
```

What is the result?

- A. It results in a compile time error at line 18.
- B. It results in a compile time error at line 9.
- C. It prints : 1
- D. It results in a compile time error at lines at lines 12 and 15.

Answer: A

Explanation:

```
5 public static void main(String[] args) {
6     char colorCode = 'y';
7     switch (colorCode) {
8         case 'r':
9             int color = 100;
10            break;
11        case 'b':
12            color = 10; // doesn't result error because color is initialized in the switch block
13            break;
14        case 'y':
15            color = 1; // doesn't result error because color is initialized in the switch block
16            break;
17    }
18    System.out.println(color); // cannot read the variable in the switch block
19 }
```

Console <terminated> Q023 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 20, 2020, 11:28:53 PM -)

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
color cannot be resolved to a variable

24. Given:

```
class A {  
    public A(){  
        System.out.print("A ");  
    }  
}  
  
class B extends A{  
    public B(){  
        System.out.print("B "); //line n1  
    }  
}  
  
class C extends B{  
    public C(){ //line n2  
        System.out.print("C ");  
    }  
    public static void main(String[] args) {  
        C c = new C();  
    }  
}
```

What is the result?

- A. C
- B
- A
- B. C
- C. A
- B
- C
- D. Compilation fails at line n1 and line n2

Answer: C

Explanation:

```
3 class A{  
4     public AO {  
5         System.out.println("A ");  
6     }  
7 }  
8  
9 class B extends A{  
10    public BO {  
11        System.out.println("B ");  
12    }  
13 }  
14  
15 class C extends B{  
16    public CO {  
17        System.out.println("C");  
18    }  
19    public static void main(String[] args) {  
20        C c = new CO;  
21    }  
22 }
```

Console ⇗
<terminated> C [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0
A
B
C

25. Given the code fragment:

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

What is the result?

A. 1:

2:

3:

4:

5:

B. 1:

2:

3:

C. Compilation fails.

D. An `ArrayOutOfBoundsException` is thrown at runtime.

Answer: B

Explanation:

```
4
5 public static void main(String[] args) {
6     int nums1[] = {1, 2, 3};
7     int nums2[] = {1, 2, 3, 4, 5};
8     nums2 = nums1;
9     for (int x : nums2) {
10         System.out.println(x + ": ");
11     }
12 }
13 }
```

Console >

<terminated> Q025 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_25

1:
2:
3:

26. Given:

```
public class FieldInit {  
    Character c;  
    boolean b;  
    float f;  
    void printAll() {  
        System.out.println("c = " + c);  
        System.out.println("b = " + b);  
        System.out.println("f = " + f);  
    }  
  
    public static void main(String[] args) {  
        FieldInit f = new FieldInit();  
        f.printAll();  
    }  
}
```

What is the result?

- A. c= null b= true f= 0.0
- B. c= b= false f= 0.0
- C. c= null b=false f= 0.0
- D. c= 0 b=false f=0.0F

Answer: C

Explanation:

```
3 public class FieldInit {  
4     Character c;  
5     boolean b;  
6     float f;  
7     void printAll() {  
8         System.out.println("c = " + c);  
9         System.out.println("b = " + b);  
10        System.out.println("f = " + f);  
11    }  
12  
13    public static void main(String[] args) {  
14        FieldInit f = new FieldInit();  
15        f.printAll();  
16    }  
17 }
```

Console >
<terminated> FieldInit [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Content

```
c = null  
b = false  
f = 0.0
```

byte = 0
short = 0
int = 0
long = 0
float = 0.0
double = 0.0
boolean = false
char =
Byte = null
Short = null
Integer = null
Long = null
Float = null
Double = null
Boolean = null
Character = null
String = null
StringBuilder = null
StringBuffer = null
String array = null
int array = null

27. Given this class:

```
public class CheckingAccount {  
    public int amount;  
    //line n1  
}
```

And given this main method, located in another class:

```
public static void main(String[] args) {  
    CheckingAccount acct = new CheckingAccount();  
    //line n2  
}
```

Which three pieces of code, when inserted independently, set the value of amount to 100?

A

```
At line n1 insert:  
public CheckingAccount() {  
    amount = 100;  
}
```

B

```
At line n2 insert:  
this.amount = 100;
```

C

```
At line n2 insert:  
amount = 100;
```

D

```
At line n1 insert:  
public CheckingAccount() {  
    this.amount = 100;  
}
```

E

```
At line n2 insert:  
acct.amount = 100;
```

F

```
At line n1 insert:  
public CheckingAccount() {  
    acct.amount = 100;  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

Answer: A D E

Explanation:

```
public class CheckingAccount {  
    public int amount;  
    // line n1|  
  
    public static void main(String[] args) {  
        CheckingAccount acc = new CheckingAccount();  
        // line n2  
    }  
}  
/*  
* Which three pieces of code, when inserted independently, set the value of amount to 100?  
*/
```

A. At line n1 insert:

```
public CheckingAccount(){  
    amount = 100;  
}
```

B. At line n2 insert:

```
this.amount = 100; // ==> Cannot use this in a static context
```

C. At line n2 insert:

```
amount = 100; // ==> Cannot make a static reference to the non-static field amount
```

D. At line n1 insert:

```
public CheckingAccount(){  
    this.amount = 100;  
}
```

E. At line n2 insert:

```
acc.amount = 100;
```

F. At line n1 insert:

```
public CheckingAccount(){  
    acc.amount = 100; // ==> acc cannot be resolved to a variable  
}
```

```
*/
```

28. 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 UnsupportedOperationException is thrown at runtime.

Answer: C

Explanation:

```
9 public static void main(String[] args) {  
10    String[] arr = {"Hi", "How", "Are", "You"};  
11    List<String> arrList = new ArrayList<>(Arrays.asList(arr));  
12    if(arrList.removeIf(s -> {System.out.print(s); return s.length() <= 2;} )){  
13        System.out.println(" removed");  
14    }  
15 }  
16 }
```

Console
<terminated> Q028 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 21, 2020)
HiHowAreYou removed

Another version:

```
8  
9 public static void main(String[] args) {  
10    String[] arr = {"Hi", "How", "Are", "You"};  
11    List<String> arrList = new ArrayList<>(Arrays.asList(arr));  
12    if(arrList.removeIf(s -> {System.out.print(s); return s.length() < 2;} )){  
13        System.out.println(" removed");  
14    }  
15 }  
16 }
```

Console
<terminated> Q028 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 21, 2020, 1)
HiHowAreYou

29. 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(1);  
    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: E

Explanation:

```
6 public static void main(String[] args) {
7     ArrayList<Integer> points = new ArrayList<>();
8     points.add(1); // adds int value to the index number 0 and so on
9     points.add(2);
10    points.add(3);
11    points.add(4);
12    points.add(null);
13    points.remove(1); // removes index!!!
14    points.remove(null); //removes object and returns boolean
15    System.out.println(points);
16
17 }
18 }
```

Console ☒

```
<terminated> Q029 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (J
[1, 3, 4]
```

30. 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

Explanation:

```
7
8 public static void main(String[] args) {
9     List<String> names = new ArrayList<>();
10    names.add("Robb");
11    names.add("Bran");
12    names.add("Rick");
13    names.add("Bran");
14
15    if (names.remove("Bran")) {
16        names.remove("Jon");
17    }
18    System.out.println(names);
19 }
20 }
```

Console ☰

```
<terminated> Q030 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Con
[Robb, Rick, Bran]
```

31. Given the code fragment:

```
int array1[] = {1, 2, 3};  
int array2[] = new int [5];  
array2 = array1;  
for (int i : array2) {  
    System.out.print(i + " ");  
}  
System.out.println();  
int array3[] = new int[3];  
array3 = array2;  
for (int i : array3) {  
    System.out.print(i + " ");  
}
```

What is the result?

- A. 1 2 3 0 0 1 2 3 0 0
- B. An Exception is thrown at run time.
- C. 1 2 3 0 0 1 2 3
- D. 1 2 3 1 2 3

Answer: D

Explanation:

```
7 int array1 [] = {1, 2, 3}; // 1 2 3
8 int array2 [] = new int[5]; // 0 0 0 0 0
9
10 array2 = array1; // 1 2 3      (not 1 2 3 null null)
11
12 for (int i : array2) {
13     System.out.print(i + " ");
14 }
15 System.out.println();
16 int array3 [] = new int[3];
17 array3 = array2; // 1 2 3
18
19 for (int i : array3) {
20     System.out.print(i + " ");
21 }
22 }
23 }
```

Console 
<terminated> Q031 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/H
1 2 3
1 2 3

32. Given:

```
class X {  
    int i;  
    static 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 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 5 6

Answer: D

Explanation:

```
2
3 public class X {
4     int i;
5     static int j;
6
7     public static void main(String[] args) {
8         X x1 = new X();
9         X x2 = new X();
10        x1.i = 3;
11        x1.j = 4;
12        x2.i = 5;
13        x2.j = 6;
14        System.out.println(
15            x1.i + " " +
16            x1.j + " " +
17            x2.i + " " +
18            x2.j);
19    }
20 }
```

Console > X [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251
3 6 5 6

33. 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 A

B. Option B

C. Option C

D. Option D

E. Option E

Answer: A C

Explanation: 'toString()' method : Object class'inda zaten oldugu icin bu methodu yazdigimizda override goruruz ve her object olusturdugumuzda bu method calisir. Aslinda override etmeyince zaten memory'deki yerini gosterir.

```
4
3 public class Employee {
4   String name;
5   boolean contract;
6   double salary;
7   Employee() {
8     //line1
9   }
10  public String toString() {
11    return name + ":" + contract + ":" + salary;
12  }
13  public static void main(String[] args) {
14    Employee e = new Employee();
15    // line2
16    e.name = "Joe";
17    e.contract = true;
18    e.salary = 100;
19    System.out.println(e);
20  }
21 }
```

Console
<terminated> Employee [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8
Joe:true:100.0

```
4
3 public class Employee {
4   String name;
5   boolean contract;
6   double salary;
7   Employee() {
8     //line1
9     this.name = new String("Joe");
10    this.contract = new Boolean(true);
11    this.salary = new Double(100);
12  }
13  public String toString() {
14    return name + ":" + contract + ":" + salary;
15  }
16  public static void main(String[] args) {
17    Employee e = new Employee();
18    // line2
19    System.out.println(e);
20  }
21 }
```

Console
<terminated> Employee [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8
Joe:true:100.0

A: Replace line2 with:

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

B: Replace line2 with:

```
this.name = "Joe";      // this. cannot be used in static content / main method
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: (was n2 in earlier versions, non-static fields can not be called under static methods)

```
name = "Joe";
contract = TRUE;      // --> TRUE gives compilation error
salary = 100.0f;
```

E: Replace line n1 with:

```
this("Joe", true, 100); // could be used to call a constructor w/ 3 parameters
```

34. Given:

```
public class Triangle {  
    static double area;  
    int b = 2, h = 3;  
    public static void main(String[] args) {  
        double p, b, h;          //line n1  
        if (area == 0) {  
            b = 3;  
            h = 4;  
            p = 0.5;  
            area = p * b * h;      //line n2  
        }  
        System.out.println("Area is " + area);  
    }  
}
```

What is the result?

- A. Area is 6.0
- B. Area is 3.0
- C. Compilation fails at line n1
- D. Compilation fails at line n2.

Answer: A

Explanation:

The screenshot shows a code editor with Java code and a terminal window below it. The code editor highlights line 8 with a green comment: // line n1 --> same variableName can be assigned to different data types ✓. The terminal window shows the output: Area is 6.0.

```
3 public class Triangle {  
4     static double area;  
5     int b= 2, h = 3;  
6  
7     public static void main(String[] args) {  
8         double p, b, h;          // line n1 --> same variableName can be assigned to different data types ✓  
9         if (area == 0) {  
10             b = 3;  
11             h = 4;  
12             p = 0.5;  
13             area = p * b * h;      // line n2  
14         }  
15         System.out.println("Area is " + area);  
16     }  
17 }
```

Console <terminated> Triangle [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 21, 2020, 12:15:43 PM - 12:1
Area is 6.0

35. 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 True Done?

- A. Replace line 5 With String opt = "true"; Replace line 7 with case "true":
- B. Replace line 5 with boolean opt = 1; Replace line 7 with case 1:
- C. At line 9, remove the break statement.
- D. Remove the default section.

Answer: A

Explanation:

switch does not work with long, double, float and boolean

```
~ 3 public class Q035 {  
4  public static void main(String[] args) {  
5      String opt = "true";  
6      switch (opt) {  
7          case "true":  
8              System.out.println("True");  
9              break;  
10         default:  
11             System.out.println("****");  
12         }  
13         System.out.println("Done");  
14     }  
15 }
```

Console

<terminated> Q035 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.

True

Done

36. 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 B C Work done
- B. A B C D Work done
- C. A Work done
- D. Compilation fails

Answer: C

Explanation:

```
5 public static void main(String[] args) {  
6     String[] arr = {"A", "B", "C", "D"};  
7     for (int i = 0; i < arr.length; i++) {  
8         System.out.println(arr[i] + " "); // prints arr[0] which is A  
9         if (arr[i].equals("C")) { // returns false for arr[0]  
10            continue; // if condition is false so does not continue in loop  
11        }  
12        System.out.println("Work Done"); // printed out  
13        break;  
14    }  
15 }  
16 }
```

Console >
<terminated> Q036 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java
A
Work Done

Other possible versions:

```
5 public static void main(String[] args) {  
6     String[] arr = {"A", "B", "C", "D"};  
7     for (int i = 0; i < arr.length; i++) {  
8         System.out.println(arr[i] + " ");  
9         if (arr[i].equals("C")) {  
10            continue;  
11        }  
12        System.out.println("Work Done");  
13        continue;  
14    }  
15 }
```

Console >
<terminated> Q36_5 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java
A
Work Done
Work Done
C
D
Work Done

```
4 public static void main(String[] args) {  
5     String[] arr = {"A", "B", "C", "D"};  
6     for (int i = 0; i < arr.length; i++) {  
7         System.out.println(arr[i] + " "); // 1. arr[0] =String "A"  
8         if (arr[i].equals("A")) { // 1. returns true  
9             break; // 1. break to loop and goes out of it  
10        }  
11        System.out.println("Work Done"); // NOT printed out  
12        continue;  
13    }  
14 }  
15 }
```

Console >
<terminated> Q36_4 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java
A

```
5 public static void main(String[] args) {  
6     String[] arr = {"A", "B", "C", "D"};  
7     for (int i = 0; i < arr.length; i++) {  
8         System.out.println(arr[i] + " "); // 1. arr[0] =String "A" - 2. arr[1] = "B"  
9         if (arr[i].equals("A")) { // 1. returns true - 2. returns false  
10            continue; // 1. continues in the loop - 2. does not continue in the loop  
11        }  
12        System.out.println("Work Done"); // printed out  
13        break;  
14    }  
15 }
```

Console >
<terminated> Q36_2 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 21, 2018)
A
B
Work Done

```
4 public static void main(String[] args) {  
5     String[] arr = {"A", "B", "C", "D"};  
6     for (int i = 0; i < arr.length; i++) {  
7         System.out.println(arr[i] + " "); // 1. arr[0] =String "A"  
8         if (arr[i].equals("A")) { // 1. returns true  
9             continue; // 1. goes back into inner loop  
10        }  
11        System.out.println("Work Done"); // printed out  
12        continue; // goes back into the loop  
13    }  
14 }  
15 }
```

Console >
<terminated> Q36_3 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java
A
B
Work Done
C
Work Done
D
Work Done

37. Given the code fragment:

```
public static void main(String[] args) {  
    int data[] = {2010, 2013, 2014, 2015, 2014};  
    int key = 2014;  
    int count = 0;  
    for (int e: data) {  
        if (e != key) {  
            continue;  
            count++;  
        }  
    }  
    System.out.print(count + " Found");  
}
```

What is the result?

- A. Compilation fails.
- B. 0 Found
- C. 1 Found
- D. 3 Found

Answer: A

Explanation:

Count; cannot be following break; or continue; statements!!

```
4
5 public static void main(String[] args) {
6     int data [] = {2010, 2013, 2014, 2015, 2014};
7     int key = 2014;
8     int count = 0;
9     for (int e : data) {
10         if (e !=key) {
11             continue; // when matched goes back in to loop
12             count++; // cannot come after continue
13         }
14     }
15     System.out.println(count + " Found");
16 }
17
18 }
```

Console ><terminated> Q037 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
 Unreachable code

Other possible versions:

```
4
5 public static void main(String[] args) {
6     int data [] = {2010, 2013, 2014, 2015, 2014};
7     int key = 2014;
8     int count = 0;
9     for (int e : data) {
10         if (e !=key) {
11             count++;
12             continue;
13         }
14     }
15     System.out.println(count + " Found");
16 }
17 }
```

Console ><terminated> Q37_2 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.
3 Found

```
4
5 public static void main(String[] args) {
6     int data [] = {2010, 2013, 2014, 2015, 2014};
7     int key = 2014;
8     int count = 0;
9     for (int e : data) {
10         if (e !=key) {
11             count++;
12             break;
13         }
14     }
15     System.out.println(count + " Found");
16 }
17 }
```

Console ><terminated> Q37_2 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.
1 Found

```
4
5 public static void main(String[] args) {
6     int data [] = {2010, 2013, 2014, 2015, 2014};
7     int key = 2014;
8     int count = 0;
9     for (int e : data) {
10         if (e !=key) {
11             continue;
12         }
13     }
14     System.out.println(count + " Found");
15     count++;
16 }
17 }
```

Console ><terminated> Q37_2 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.
0 Found

38. Which three are advantages of the Java exception mechanism? (Choose three.)

- A. Improves the program structure because the error handling code is separated from the normal program function ✓
- B. Provides a set of standard exceptions that covers all possible errors ✗
- C. Improves the program structure because the programmer can choose where to handle exceptions ✓
- D. Improves the program structure because exceptions must be handled in the method in which they occurred Text
- E. Allows the creation of new exceptions that are customized to the particular program being created ✓

Answer: **A C E**

Explanation:

39. 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. System.out.println(tiger.type + " " + tiger.maxSpeed + " " + tiger.bounds);
```

and this output:

Canine 60 Long

Feline 80 Short

Which two modifications enable the code to print this output? (Choose two.)

- A Replace line n1 with:

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

- B Replace line n1 with:

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

- C Replace line n2 with:

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

- D Replace line n1 with:

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

- E Replace line n2 with:

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

Answer: A E

Explanation:

super() is used to call Base class's(Parent class's) constructor.
this() is used to call **current class's constructor**.

```
3  class Animal{
4      String type = "Canine";
5      int maxSpeed = 60;
6
7      Animal() {}
8
9      Animal (String type, int maxSpeed) {
10         this.type = type;
11         this.maxSpeed = maxSpeed;
12     }
13 }
14
15 class WildAnimal extends Animal{
16     String bounds;
17
18     WildAnimal (String bounds) {
19         // line n1
20
21         super(); // calls fields via non-parameter constructor of parent class ** if constructor is not created gives compilation error **
22         this.bounds = bounds;
23     }
24
25     WildAnimal(String type, int maxSpeed, String bounds) {
26         super(type, maxSpeed); // line n2
27         this.bounds = bounds;
28     }
29 }
30 public static void main(String[] args) {
31     WildAnimal wolf = new WildAnimal("Long");
32     WildAnimal tiger = new WildAnimal("Feline", 80, "Short");
33     System.out.println(wolf.type + " " + wolf.maxSpeed + " " + wolf.bounds);
34     System.out.println(tiger.type + " " + tiger.maxSpeed + " " + tiger.bounds);
35 }
36 }
```

Console <terminated> WildAnimal [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java (Jun 21, 2020, 1:30:13 PM – 1:30:13 PM)
Canine 60 Long
Feline 80 Short

B. Replace line n1 with:

```
this.bounds = bounds;  
super (); // super() and this() should be the first statement following constructor --> compilation error
```

C. Replace line n2 with:

```
super(type, maxSpeed);  
this (bounds); // super() and this() should be the first statement following constructor --> compilation error
```

D. Replace line n1 with:

```
this("Canine", 60); // should've been defined as this -- ("Canine", 60, bounds);-- to call current class's 3 parameter constructor  
this.bounds = bounds;
```

40. 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[] keys = findMax(numbers);
    }

    /* line n1 */
    int[] keys = new int[3];
    /* code goes here*/
    return keys;
}
```

Which method signature do you use at line n1?

- A. public int findMax (int[] numbers)
- B. static int[] findMax (int[] max)
- C. static int findMax (int[] numbers)
- D. final int findMax (int[])

Answer: B

Explanation:

```
public static void main(String[] args) {  
    int [] numbers = {12, 13, 42, 32, 15, 156, 23, 51,12};  
    int [] keys = findMax(numbers);  
}  
/* line n1 */static int[]findMax (int[] max) {  
    int[] keys = new int [3];  
    /* code goes here*/  
    return keys;  
}  
}
```

41. 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 A

B. Option B

C. Option C

D. Option D

Answer: D

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

- A. sb. deleteAll ();
- B. sb. delete (0, sb. size ());
- C. sb. delete (0, sb. length ());
- D. sb. removeAll ();

Answer: C

43. 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. Compilation fails.
- B. Hi removed
- C. An UnsupportedOperationException is thrown at runtime.
- D. The program compiles, but it prints nothing.

Answer: A

44. Given the code fragment:

```
public class Game {  
    public static void menu() {  
        system.out.println("1. Left 2. Right 0. Stop");  
    }  
    public static void main(String[] args) {  
        int option;  
        /* insert code here */  
    }  
}
```

and the requirements of the application:

- It must display the menu.
- It must print the option selected.
- It must continue its execution till it reads '0'.

Which code fragment can be used to meet the requirements?

- A.

```
for (option = 0; option != 0; option = //code that reads the option  
goes here) {  
    /* code that print the option go here */  
}
```
- B.

```
while (option != 0) {  
    menu();  
    option = // code that reads the option goes here  
    /* code that print the option go here */  
}
```
- C.

```
do {  
    menu();  
    option = // code that reads the option goes here  
    /* code that print the option go here */  
} while (option != 0);
```
- D.

```
while (option >= 0) {  
    menu ();  
    option = // code that reads the option goes here  
    /* code that print the option go here */  
}
```

A. Option A

B. Option B

C. Option C

D. OptionD

Answer: ~~A~~ < Ans=c

45. 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 n1 public.
- B. Make the method at line n2 public.
- C. Make the method at line n3 public. ✓
- D. Make the method at line n3 protected. ✓
- E. Make the method at line n4 public. ✗

Answer: C D

46. Given the code fragment:

```
int a = 3;  
int b = 2;  
int c = 1;  
int r1 = a * b / c + 1; 7  
int r2 = a / b * c + 1; 2.5 then 2  
int r3 = a * (b / (c + 1));  
System.out.println(r1 + " : " + r2 + " : " + r3);
```

What is the result?

- A. 2 : 7 : 3
- B. 7 : 7 : 9
- C. 2 : 7 : 0
- D. 7 : 2 : 3

Answer: D

Explanation::

Result

CPU Time: 0.32 sec(s), Memory: 35824 kilobyte(s)

7 : 2 : 3

47. Given the class definitions:

```
class C1 {}  
class C2 extends C1 {}  
class C3 extends C2 {}
```

and the code fragment:

- * 16. C1 obj1 = (C1) new C2();
- * 17. C2 obj2 = (C2) new C3();
- * 18. C2 obj3 = (C2) new C1();
- * 19. C3 obj4 = (C3) obj2;

Which line throws ClassCastException?

- A. line 18
- B. line 17
- C. line 19
- D. line 16

Answer: D

Explanation::

```
Exception in thread "main" java.lang.ClassCastException: class CC$1C1 cannot be cast to class CC$1C2 (CC$1C1 and CC$1C2 are in unnamed module of loader 'app'  
at CC.main(CC.java:9)
```

48. Given:

```
public class Test {  
    // line n1  
}
```

Which two code fragments can be inserted at line n1?

- A. String str = "Java";
- B. for(int iVal = 0; iVal <=5; iVal++){} 
- C. Test() {} 
- D. package p1; 
- E. import java.io.*; 

Answer: A D

49. Given the code fragment:

```
int n [] [] = {{1, 3}, {2, 4}};
for (int i = n.length-1; i >= 0; i--) {
    for (int y : n[i]) {
        System.out.print (y);
    }
}
```

What is the result?

- A. 1324
- B. 2313
- C. 3142
- D. 4231

Answer: D

50. Which two code fragments cause compilation errors? (Choose two.)

A. double y1 = 203.22; float fit ~~= y1;~~

B. float fit = (float) 1_11.00;

C. Float fit = 100.00; D. int y2 = 100;

float fit = (float) y2;

E. float fit = 100.00F;

Answer: B D

51. Given the code fragments:

```
Interface Exportable {  
    Void export();  
}  
  
class Tool implements Exportable {  
    protected void export () {          //line n1  
        System.out.println("Tool::export");  
    }  
}  
  
class ReportTool extends Tool implements Exportable {  
  
    public 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 n2.
- B. RTool::exportTool::export
- C. Tool::exportTool:export
- D. Compilation fails only at line n1.
- E. Compilation fails at both line n1 and line n2.

Answer: E

52. 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. 97 9899 100 null null null
- B. 97 9899 100 101 102 103
- C. **Compilation fails.**
- D. **A NullPointerException is thrown at runtime.**
- E. **An ArrayIndexOutOfBoundsException is thrown at runtime.**

Answer: A

53. 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 B
- C. Option C
- D. Option D

Answer: A

54. You are asked to develop a program for a shopping application, and you are given this 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 B

C. Option C

D. Option D

Answer: A

55. 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. }  
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 6 with System.out.print(--x);
- B. At line 7, insert x --;
- C. Replace line 6 with --x; and, at line 7, insert System.out.print(x);
- D. Replace line 12 with return (x > 0) ? false: true;

Answer: B

56. 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. true false
- C. false false
- D. false true

Answer: C

57. 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 A

B. Option B

C. Option C

D. Option D

Answer: C

58. Which statement is true about the `main()` method?

- A. It is invoked by JRE
- B. It is a final method
- C. It returns true if it is executed successfully at run time
- D. It must be defined within a public class

Answer: A

59. 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 A

B. Option B

C. Option C

D. Option D

E. Option E

Answer: A C

60. 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 A

B. Option B

C. Option C

D. Option D

Answer: D

61. 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 reverse order of entry.
- * 2. Process all the elements of the array in the order of entry.
- * 3. Process alternating elements of the array in the order of entry.
- * Which two statements are true? (Choose two.)

- A. Requirements 1, 2, and 3 can be implemented by using the enhanced for loop.
- B. Requirements 1, 2, and 3 can be implemented by using the standard for loop.
- C. Requirements 2 and 3 CANNOT be implemented by using the standard for loop.
- D. Requirement 2 can be implemented by using the enhanced for loop.
- E. Requirement 3 CANNOT be implemented by using either the enhanced for loop or the standard for loop.

Answer: B C

62. Given:

```
interface Downloadable {  
    public void download();  
}  
  
interface Readable extends Downloadable {      // line n1  
    public void readBook();  
}  
  
abstract class Book implements Readable {        // line n2  
    public void readBook() {  
        System.out.println("Read Book");  
    }  
}  
  
class EBook extends Book {                      // line n3  
    public void readBook() {  
        System.out.println("Read E-Book");  
    }  
}
```

And given the code fragment:

```
Book book1 = new EBook();  
book1.readBook();
```

What is the result?

- A. Compilation fails at line n2.
- B. Read Book
- C. Read E-Book
- D. Compilation fails at line n1.
- E. Compilation fails at line n3.

Answer: E

63. 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. Compilation fails only at line n2.
- C. Tool::exportTool::export
- D. Compilation fails at both line n1 and line2.
- E. RTool::exportTool::export

Answer: A

64. Given the definitions of the MyString class and the Test class:

MyString.java:

```
package p1;
class MyString {
    String msg;
    MyString(String msg) {
        this.msg = msg;
    }
}
```

Test.java:

```
package p1;
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello " + new StringBuilder("Java SE 8"));
        System.out.println("Hello " + new MyString("Java SE 8"));
    }
}
```

What is the result?

A

```
Hello Java SE 8
Hello Java SE 8
```

B

```
Hello java.lang.StringBuilder@<<hashcode1>>
Hello p1.MyString@<<hashcode2>>
```

C

```
Hello Java SE 8
Hello p1.MyString@<<hashcode>>
```

D Compilation fails at the Test class

A. Option A

B. Option B

C. Option C

D. Option D

Answer: C

65. Given the code fragment:

```
24. float vari = (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. An exception is thrown at runtime.

B. Compilation fails.

C. 13480.0

D. 13480.02

Answer: C

66. Given:

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

And the code fragment:

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

```
C2 s = (C2) obj2;  
I t = obj1;
```

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

What is the result?

A. C1C2

B. C1C1

C. Compilation fails.

D. C2C2

Answer: A

Explanation::

Blund

src

```
App.java
1
2 interface I {
3     public void displayI();
4 }
5 abstract class C2 implements I {
6     public void displayC2() {
7         System.out.print("C2");
8     }
9 }
10 class C1 extends C2 {
11     public void displayI() {
12         System.out.print("C1");
13     }
14 }
15 }
16
17 public class App {
18     public static void main(String[] args) {
19         C2 obj1 = new C1();
20         I obj2 = new C1();
21
22         C2 s = (C2) obj2;
23         I t = obj1;
24
25         t.displayI();
26         s.displayC2();
27     }
28 }
29 }
```

Console 1

C1C2

Completed with exit code: 0

67. Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start();  
        c.init();  
    }  
}
```

What is the result?

- A. An exception is thrown at runtime.
- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails.

Answer: D

68. Given this code for the classes MyException and Test:

```
public class MyException extends RuntimeException {}

public class Test {
    public static void main(String[] args) {
        try {
            method1();
        }
        catch (MyException ne) {
            System.out.print("A");
        }
    }
    public static void method1() { // line n1
        try {
            throw 3 > 10 ? new MyException() : new IOException();
        }
        catch(IOException ie) {
            System.out.println("I");
        }
        catch (Exception re) {
            System.out.print("B");
        }
    }
}
```

What is the result?

- A. A
- B. AB
- C. A compile time error occurs at line n1.
- D. B
- E. I

Answer: C

Explanation::

```

1
2 public class Test {
3     public static void main(String[] args){
4         try {
5             method1();
6         }
7         catch (MyException ne) {
8             System.out.println("A");
9             illegal start of expression
10    }
11    public static void method1() {
12        try {
13            throw 3 > 10 ? new MyException() : new IOException
14        }
15        catch(IOException ie) {
16            System.out.println("I");
17        }
18        catch (Exception re) {
19            System.out.print("B");
20        }
21    }

```

69. Given:

```

public class Test {

    public static void main(String[] args) {

        String[][] chs = new String[5][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. 97 9899 100 null null null
- B. 97 9899 100 101 102 103
- C. Compilation fails.
- D. A NullPointerException is thrown at runtime.
- E. An ArrayIndexOutOfBoundsException is thrown at runtime.

Answer: E

Explanation::

```
Console 8 ✘ | Console 9 ✘
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 2 out of bounds for length 2
    at Test.main(Test.java:11)

Completed with exit code: 1
```

70. Given the code fragment:

```
int x = 10;
int y = ++x;
int z = 0;
if (y >= 10 || y <= ++x) {
    z = x;
} else {
    z = x++;
}
System.out.println(z);
```

What is the result?

- A. 11
- B. 10 C. 12
- D. A compile time error occurs.

Answer: C

Explanation::

Result

CPU Time: 0.14 sec(s), Memory: 32028 kilobyte(s)

12

71. 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. C2C2
- B. C1C2
- C. C1C1
- D. Compilation fails

Answer: B

72. 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 true

C. true false

D. true true

E. false false

Answer: B 73. 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 DateTimeException is thrown at runtime.

Answer: B

Explanation::

The screenshot shows a Java IDE interface with two panes. On the left, the code for `Main.java` is displayed:

```
Main.java ① saved  
1 import java.time.LocalDate;  
2 import java.time.Month;  
3  
4 public class Main {  
5     public static void main(String[] args) {  
6         LocalDate date = LocalDate.of(2012, 1, 30);  
7         date.plusDays(10);  
8         System.out.println(date);  
9     }  
10 }
```

On the right, a terminal window shows the execution of the code:

```
java version "1.8.0_31"  
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-core-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-core-1.3.jar:/run_dir/json-simple-1.1.1.jar Main  
2012-01-30
```

74. Given the code fragment:

```
public class Test{  
  
    void readCard(int cardNo) throws Exception {  
        System.out.println("Reading Card");  
    }  
  
    void checkCard(int cardNo) throws RuntimeException { // line n1  
        System.out.println("Checking Card");  
    }  
  
    public static void main(String[] args) {  
        Test ex = new Test();  
        int cardNo = 12344;  
        ex.readCard(cardNo); //line n2  
        ex.checkCard(cardNo); //line n3  
    }  
}
```

What is the result?

A

Reading Card
Checking Card

- B Compilation fails only at line n1.
- C Compilation fails only at line n2.
- D Compilation fails only at line n3.
- E Compilation fails at both line n2 and line n3.

A. Option A

B. Option B

C. Option C

D. Option D E. Option E

Answer: D

75. Given the code fragment:

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

What is the result?

- A. 2 4
- B. 0 2 4 6
- C. 0 2 4
- D. Compilation fails.

Answer: B

Explanation::

```
Console 2  
0 2 4 6  
Completed with exit code: 0
```

76. Given the code fragment:

```
1. abstract class Planet {  
2.     protected void revolve() {  
3.     }  
4.     abstract void rotate();  
5. }  
6.  
7. class Earth extends Planet {  
8.     private void revolve() {  
9.     }  
10.    private void rotate() {  
11.    }  
12. }
```

Which two modifications enable the code to compile? A.

- Make the method at line 8 protected.
- B. Make the method at line 8 public.
- C. Make the method at line 10 protected.
- D. Make the method at line 4 public.
- E. Make the method at line 2 public.

Answer: A C

77. Given this 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 A

B. Option B

C. Option C

D. Option D

E. Option E

F. Option F

Answer: B E

78. 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.equals(str3));
        boolean b2 = (str1== str3);
        System.out.print (b1+", "+b2);
    }
}
```

What is the result?

- A. false, false
- B. false, true
- C. true, false
- D. true, true

Answer: B

79. Given:

```
class C {
    public C(){
        System.out.print("C ");
    }
}

class B extends C{
    public B() { //line n1
        System.out.print("B ");
    }
}

public class A extends B{

    public A(){ //line n2
        System.out.print("A ");
    }
    public static void main(String[] args) {
        A a = new A();
    }
}
```

What is the result?

- A. C B A
- B. C
- C. A B C
- D. Compilation fails at line n1 and line n2

Answer: A

80. Which two features can be implemented in a Java application by encapsulating the entity classes used?

(Choose two.)

- A. data validation
- B. compile time polymorphism
- C. data hiding
- D. data abstraction
- E. data memory optimization

Answer: C D

81. Which two code fragments cause a compilation error? (Choose two.)

- A. float flt = 100.00F;
- B. float flt = (float) 1_11.00;
- C. Float flt = 100.00;
- D. double y1 = 203.22;float flt = y1;
- E. int y2 = 100;float flt = (float) y2 ;

Answer: A D

82. Given:

```

class Alpha {
    int ns;
    static int s;
    Alpha(int ns) {
        if (s < ns) {
            s = ns;
            this.ns = ns;
        }
    }
    void doPrint() {
        System.out.println("ns = " + ns + " s = " + s);
    }
}

```

And,

```

public class TestA {
    public static void main(String[] args) {
        Alpha ref1 = new Alpha(50);
        Alpha ref2 = new Alpha(125);
        Alpha ref3 = new Alpha(100);
        ref1.doPrint();
        ref2.doPrint();
        ref3.doPrint();
    }
}

```

What is the result?

- A) ns = 50 s = 125
ns = 125 s = 125
ns = 100 s = 125
- B) ns = 50 s = 125
ns = 125 s = 125
ns = 0 s = 125
- C) ns = 50 s = 50
ns = 125 s = 125
ns = 100 s = 100
- D) ns = 50 s = 50
ns = 125 s = 125
ns = 0 s = 125

A. Option A

B. Option B

C. Option C

D. Option D

Answer: B

83. 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 = true;  
}
```

What is the result?

- A. Compilation fails.
- B. false true
- C. true false
- D. true true
- E. false false

Answer: C

Explanation::

```
Console 15 ✘ true false  
Console 16 ✘ Completed with exit code: 0
```

84. Which two array initialization statements are valid? (Choose two.)

- A. int array[] = new int[3] {1, 2, 3};
- B. int array[] = new int[3]; array[0] = 1; array[1] = 2; array[2] = 3;
- C. int array[3] = new int[] {1, 2, 3};
- D. int array[] = new int[3]; array = {1, 2, 3};
- E. int array[] = new int[] {1,2,3};

Answer: B E

85. Given the following code for the classes MyException and Test:

```
public class MyException extends RuntimeException {}

public class Test {
    public static void main(String[] args) {
        try {
            method1();
        }
        catch (MyException ne) {
            System.out.print("A");
        }
    }
    public static void method1() { // line n1
        try {
            throw Math.random() > 0.5 ?new MyException() :new RuntimeException();
        }
        catch (RuntimeException re) {
            System.out.print("B");
        }
    }
}
```

What is the result?

- A. A
- B. B
- C. Either A or B
- D. A B
- E. A compile time error occurs at line n1

Answer: B

86. 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();
    A b3 = (B) b2;           //line n1
    b1 = (A) b2;            //line n2
    b1.test();
    b3.test();
}
```

What is the result?

- A. AB
- B. AC
- C. CC
- D. A ClassCastException is thrown only at line n1.
- E. A ClassCastException is thrown only at line n2.

Answer: D

87. Given:

```
public class MarkList {  
    int num;  
    public static void graceMarks(MarkList obj4) {  
        obj4.num += 10;  
    }  
    public static void main(String[] args) {  
        MarkList obj1 = new MarkList();  
        MarkList obj2 = obj1;  
        MarkList obj3 = null;  
        obj2.num = 60;  
        graceMarks(obj2);  
    }  
}
```

How many MarkList instances are created in memory at runtime?

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

Answer: A

88. Given the code fragment:

```
3. public static void main(String[] args) {  
4.     int x = 6;  
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 6 with System.out.print (--x);
- B. At line 7, insert x --;
- C. Replace line 5 with while (is Available(--x)) {
- D. Replace line 12 with return (x > 0) ? false : true;

Answer: C

89. 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

Explanation:

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

90. 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. Sum is 600
- B. Compilation fails at line n1.
- C. Compilation fails at line n2.
- D. A ClassCastException is thrown at line n1.
- E. A ClassCastException is thrown at line n2.

Answer: C

91. 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 A
B. Option B
C. Option C
D. Option D E. Option E

Answer: E

92. Given the code fragment:

```
public class Test {  
    public static void main(String[] args) {  
        //line n1  
        switch (x) {  
            case 1:  
                System.out.println("One");  
                break;  
            case 2:  
                System.out.println("Two");  
                break;  
        }  
    }  
}
```

Which three code fragments can be independently inserted at line n1 to enable the code to print One?

(Choose three.)

- A. `byte x = 1;`
B. `short x = 1;`
C. `String x = "1";`
D. `long x = 1;`
E. `double x = 1;`

F. Integer x = new Integer("1");

Answer: A B F

93. Given:

```
public class Product {  
    int id;  
    String name;  
    public Product(int id, String name) {  
        this.id = id;  
        this.name = name;  
    }  
}
```

And given the code fragment:

```
4. Product p1 = new Product(101, "Pen");  
5. Product p2 = new Product(101, "Pen");  
6. Product p3 = p1;  
7. boolean ans1 = p1 == p2;  
8. boolean ans2 = p1.name.equals(p2.name);  
9. System.out.print(ans1 + ":" + ans2);
```

What is the result?

- A. true:true
- B. true:false
- C. false:true
- D. false:false

Answer: C

94. Given the code fragment:

```
public class Person {  
    String name;  
    int age = 25;  
  
    Person(String name) { // line n1  
        setName(name);  
    }  
  
    public Person(String name, int age) { // line n2  
        Person(name);  
        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. Compilation fails at both line n1 and line n2.
- B. Compilation fails only at line n2.
- C. Compilation fails only at line n1.
- D. Jesse 25Walter 52

Answer: A

95. Given:

```
class Product {  
    int id;  
    String name;  
    Product (int id, String name) {  
        this.id = id;  
        this.name = name;  
    }  
}  
public class Shop {  
    public static void main(String[] args) {  
        List<Product> lst = new ArrayList<>();  
        lst.add(new Product(10, "IceCream"));  
        lst.add(new Product(11, "Chocolate"));  
        Product p1 = new Product(10, "IceCream");  
        System.out.println(lst.indexOf(p1));  
    }  
}
```

What is the result?

- A. true
- B. false
- C. -1 D. 0

Answer: C

96. Given:

```
4. public class Shop{  
5.     public static void main(String[] args) {  
6.         int price = 1000;  
7.         int qty = 2;  
8.         String grade = "2";  
9.         double discount = 0.0;  
10.        switch(grade) {  
11.            case "1":  
12.                discount = price * 0.1;  
13.                break;  
14.            case "2":  
15.                discount = price * 0.5;  
16.                continue:  
17.            default:  
18.                System.out.println("Thank You!");  
19.        }  
20.        System.out.println(discount);  
21.    }  
22. }
```

Which statement is true?

- A. The program executes and prints: 500.0
- B. Commenting line 16 enables the program to print: Thank You! 500.0
- C. Commenting line 13 enables the program to print: Thank You! 500.0
- D. The program executes and prints: Thank You! 500.0

Answer: B

Explanation:

```

16 public class Shop {
17     public static void main(String[] args) {
18         int price = 1000;
19         int qty = 2;
20         String grade = "Z";
21         double discount = 0.0;
22         switch(grade) {
23             case "1":
24                 discount = price * 0.1;
25                 break;
26             case "2":
27                 discount = price * 0.5;
28                 //continue:
29             default:
30                 System.out.println("Thank You!");
31         }
32         System.out.println(discount);
33     }
34 }
```

Result

CPU Time: 0.16 sec(s), Memory: 32260 kilobyte(s)



```

Thank You!
500.0

```

97. Given the code fragment:

```

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

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

What is the result?

- A. Match 1
- B. Match 2
- C. No Match
- D. A NullPointerException is thrown at runtime.

Answer: A

98. 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 1 3 2 2
- B. java MyFile 2 2 2
- C. java MyFile 1 2 2 3 4
- D. java MyFile 0 1 2 3

Answer: A

99. Given the code fragment:

```

class Employee {
    private String name;
    private int age;
    private int salary;

    public Employee (String name, int age) {
        setName (name)
        setAge (age)
        setSalary (2000);
    }
    public Employee (String name, int age, int salary) {
        setSalary (salary);
        this (name, age);
    }
    //getter and setter methods for attributes go here
    public void printDetails () {
        System.out.println (name + " : " + age + " : " + salary);
    }
}

```

Test.java:

```
class Test {  
    public static void main(String[] args) {  
        Employee e1 = new Employee();  
        Employee e2 = new Employee("Jack", 50);  
        Employee e3 = new Employee("Chloe", 40, 5000);  
  
        e1.printDetails();  
        e2.printDetails();  
        e3.printDetails();  
    }  
}
```

Which is the result?

A Compilation fails in the Employee class.

B

```
null : 0: 0  
Jack : 50 : 0  
Chloe : 40 : 5000
```

C

```
null : 0 : 0  
Jack : 50 : 2000  
Chloe : 40 : 5000
```

D Compilation fails in the Test class.

E Both the Employee class and the Test class fail to compile.

A. Option A

B. Option B

C. Option C

D. Option D E. Option E

Answer: E

100. 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. Both p and s are accessible via obj.
- B. Only s is accessible via obj.
- C. Both r and s are accessible via obj.
- D. p, r, and s are accessible via obj.

Answer: B

101. 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. 10 20 30 40
- B. 0 0 30 40

- C. Compilation fails.
- D. An exception is thrown at runtime.

Answer: C 102. Given the following

code:

```
public static void main(String[] args){  
    String[] planets = {"Mercury", "Venus", "Earth", "Mars"};  
  
    System.out.println(planets.length);  
    System.out.println(planets[1].length());  
}
```

What is the output?

- A. 44
- B. 35
- C. 47
- D. 54
- E. 45
- F. 421

Answer: E

103. Given the code fragment:

```
8. public static void main(String[] args) {  
9.     int x;  
10.    /* insert code here */  
11. }
```

Which code fragment at line 10 prints Welcome 100?

```
A. for (x = 0; x < 100; ++x) {  
    System.out.println("Welcome " + x);  
}  
B. for (x = 100; x <= 100; x++) {  
    System.out.println("Welcome " + x);  
}  
C. x = 100;  
while (x <= 100) {  
    x++;  
    System.out.println("Welcome " + x);  
}  
D. x = 100;  
do {  
    ++x;  
    System.out.println("Welcome " + x);  
} while (x < 100);
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: B

104. Given:

Base.java:

```
class Base {  
    public void test(){  
        System.out.println("Base ");  
    }  
}
```

DerivedA.java:

```
class DerivedA extends Base {  
    public void test(){  
        System.out.println("DerivedA ");  
    }  
}
```

DerivedB.java:

```
class DerivedB extends DerivedA {  
    public void test(){  
        System.out.println("DerivedB ");  
    }  
    public static void main(String[] args) {  
        Base b1 = new DerivedB();  
        Base b2 = new DerivedA();  
        Base b3 = new DerivedB();  
        Base b4 = b3;  
        b1 = (Base) b2;  
        b1.test();  
        b4.test();  
    }  
}
```

What is the result?

- A. BaseDerivedA
- B. BaseDerivedB
- C. DerivedBDerivedB
- D. DerivedBDerivedA
- E. A ClassCastException is thrown at runtime.

Answer: D

105. 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. 2012-02-10
- B. 2012-02-11
- C. Compilation fails
- D. A DateTimeException is thrown at runtime.

Answer: D

106. Which two statements are true? (Choose two.) A.

- Error class is unextendable.
- B. Error class is extendable.
- C. Error is a RuntimeException.
- D. Error is an Exception.
- E. Error is a Throwable.

Answer: B C

107. 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 1 null
```

What is the result?

- A. 1 null
- B. true false
- C. false false

D. true true

E. A ClassCastException is thrown at runtime.

Answer: D

108. 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("D")) {  
            System.out.println("Work done");  
            break;  
        }  
        continue;  
    }  
}
```

What is the result?

A. A B C Work done

B. A B C D Work done

C. A Work done

D. Compilation fails

Answer: B

Explanation:

```
1 public class Main {  
2     public static void main(String[] args) {  
3         String[] arr = {"A", "B", "C", "D"};  
4         for (int i = 0; i < arr.length; i++) {  
5             System.out.print (arr[i]+ " ");  
6             if (arr[i].equals("D")) {  
7                 System.out.println("work done");  
8                 break;  
9             }  
10            continue;  
11        }  
12    }  
13 }  
14 }
```



The screenshot shows a terminal window with the following text:
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-core-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-core-1.3.jar:/run_dir/json-simple-1.1.1.jar Main
A B C D work done

109. Given these two classes:

```
public class Customer {  
    ElectricAccount acct = new ElectricAccount();  
  
    public void useElectricity(double kWh) {  
        acct.addKWh(kWh);  
    }  
}  
  
public class ElectricAccount {  
    private double kWh;  
    private double rate = 0.07;  
    private double bill;  
  
    //line n1  
}
```

Any amount of electricity used by a customer (represented by an instance of the Customer class) must contribute to the customer's bill (represented by the member variable bill) through the useElectricity method. An instance of the Customer class should never be able to tamper with or decrease the value of the member variable bill.

How should you write methods in the ElectricAccount class at line n1 so that the member variable bill is always equal to the value of the member variable kwh multiplied by the member variable rate?

A

```
public void addKWh(double kWh) {
    this.kWh += kWh;
    this.bill = this.kWh*this.rate;
}
```

B

```
public void addKWh(double kWh) {
    if (kWh > 0){
        this.kWh += kWh;
        this.bill = this.kWh * this.rate;
    }
}
```

C

```
private void addKWh(double kWh) {
    if (kWh > 0) {
        this.kWh += kWh;
        this.bill = this.kWh*this.rate;
    }
}
```

D

```
public void addKWh(double kWh) {
    if(kWh > 0) {
        this.kWh += kWh;
        setBill(this.kWh);
    }
}
public void setBill(double kWh) {
    bill = kWh*rate;
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

110. Given:

Base.java:

```
class Base {  
    public void test(){  
        System.out.println("Base ");  
    }  
}
```

DerivedA.java:

```
class DerivedA extends Base {  
    public void test(){  
        System.out.println("DerivedA ");  
    }  
}
```

DerivedB.java:

```
class DerivedB extends DerivedA {  
    public void test(){  
        System.out.println("DerivedB ");  
    }  
    public static void main(String[] args) {  
        Base b1 = new DerivedB();  
        Base b2 = new DerivedA();  
        Base b3 = new DerivedB();  
        b1 = (Base) b3;  
        Base b4 = (DerivedA) b3;  
        b1.test();  
        b4.test();  
    }  
}
```

What is the result?

- A. BaseDerivedA
- B. BaseDerivedB
- C. DerivedBDerivedB
- D. DerivedBDerivedA
- E. A ClassCastException is thrown at runtime.

Answer: C

111. 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. 2 4
- B. 0 2 4 6
- C. 0 2 4
- D. Compilation fails

Answer: C

112. 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 n3 and line n4.
- B. Compilation fails at line n1 and line n2.
- C. Welcome Visit Count:1Welcome Visit Count: 1
- D. Welcome Visit Count:1Welcome Visit Count: 2

Answer: B

113. Given the code fragment:

```

public class StockRoom {
    private int stock = 10;
    public void purchase(int qty) {stock += qty;}
    public void sell(int qty) {stock -= qty}
    public void printStock(String action) {
        System.out.println(action + ":" + qty + " items. Stock
in Hand: " + stock);
    }
    public static void main(String[] args) {
        StockRoom k1 = new StockRoom();
        k1.sell(5);
        k1.printStock("Sold");
        StockRoom k2 = new StockRoom();
        k2.purchase(5);
        k2.printStock("Purchased");
    }
}

```

You want the code to print:

Sold: 5 items. Stock in Hand: 5 Purchased: 5 items. Stock in Hand: 10? Which action enables the code to print this?

- A. Declare the stock variable and the purchase(), sell(), and printStock() methods static.
- B. Declare the stock variable and the printStock() method static.
- C. Declare the stock and qty variables and the printStock() method static.
- D. Declare the stock variable static.

Answer: C

114. 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. A B C

B. A B C D E

C. A B D E

D. Compilation fails.

Answer: C

115. Given the code fragment:

```
LocalDate date1 = LocalDate.now();
LocalDate date2 = LocalDate.of(6, 20, 2014 );
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
System.out.println("date1 = " + date1);
System.out.println("date2 = " + date2);
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?

A

```
date1 = 2014-06-20
date2 = 2014-06-20
date3 = 2014-06-20
```

B

```
date1 = 06/20/2014
date2 = 2014-06-20
date3 = Jun 20, 2014
```

C Compilation fails.

D An exception is thrown at runtime.

A. Option A

B. Option B

C. Option C

D. Option D

Answer: A

116. 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. AB
- B. AC
- C. CC
- D. A ClassCastException is thrown only at line n1.
- E. A ClassCastException is thrown only at line n2.

Answer: B

117. 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 C D
- C. A C D D
- D. A B D
- E. A B D C

Answer: C

118. Given the code fragment:

```
public class App {  
    public static void main(String[] args) {  
        String str1 = "Java";  
        String str2 = new String("java");  
        //line n1  
        {  
            System.out.println("Equal");  
        } else {  
            System.out.println("Not Equal");  
        }  
    }  
}
```

Which code fragment, when inserted at line n1, enables the App class to print Equal?

- A) Str1.toLowerCase();
if (str1 == str2)
 - B) if (str2.equals(str1.toLowerCase()))
 - C) Str1.toLowerCase();
if (str1.equals(str2))
 - D) if (str1.toLowerCase() == str2.toLowerCase())
- A. Option A
B. Option B
C. Option C
D. Option D

Answer: B

119. Given:

```

public class App {
    String greet = "Welcome!";
    public App() {
        String greet = "Hello!";
    }
    public void setGreet() {
        String greet = "Good Day!";
    }

    public static void main (String[] args) {
        App t = new App();
        String greet = "Good Luck!";
        System.out.println(t.greet);
    }
}

```

What is the result? A.

Good Luck!

B. Good Day!

C. Welcome!

D. Hello!

Answer: C

120. Given:

```

class Cart {
    Product p;
    double totalAmount;
}

class Product {
    String name;
    Double price;
}

public class Shop {
    public static void main(String[] args) {
        Cart c = new Cart();
        System.out.println(c.p + ":" + c.totalAmount);
    }
}

```

What is the result?

- A. null:null:0.0
- B. null:null
- C. <>HashCode>>:0.0
- D. null:0.0

Answer: D

121. Given:

MainTest.java:

```
public class MainTest {  
  
    public static void main(int[] args) {  
        System.out.println("int main " + args[0]);  
    }  
    public static void main(Object[] args) {  
        System.out.println("Object main " + args[0]);  
    }  
    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. int main 1
- B. Object main 1
- C. String main 1
- D. Compilation fails
- E. An exception is thrown at runtime

Answer: C 122. Given the code

fragment:

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

What is the result?

- A. Sum is 600
- B. Compilation fails at line n1.
- C. Compilation fails at line n2.
- D. A ClassCastException is thrown at line n1.
- E. A ClassCastException is thrown at line n2.

Answer: E

123. Given:

```
public class App {  
  
    String myStr = "9009";  
  
    public void doStuff(String str) {  
        int myNum = 0;  
        try {  
            String myStr = str;  
            myNum = Integer.parseInt(myStr);  
        } catch (NumberFormatException ne) {  
            System.err.println("Error");  
        }  
        System.out.println(  
            "myStr: " + myStr + ", myNum: " + myNum);  
    }  
  
    public static void main(String[] args) {  
        App obj = new App();  
        obj.doStuff("7007");  
    }  
}
```

What is the result?

- A. myStr: 7007, myNum: 7007
- B. Error
- C. myStr: 9009, myNum: 7007
- D. myStr: 7007, myNum: 9009

Answer: C

Explanation:

Result

CPU Time: 0.30 sec(s), Memory: 35792 kilobyte(s)

```
myStr: 9009, myNum: 7007
```

124. Given:

```
public class App {  
  
    String myStr = "7007";  
  
    public void doStuff(String str) {  
        int myNum = 0;  
        try {  
            String myStr = str;  
            myNum = Integer.parseInt(myStr);  
        } catch (NumberFormatException ne) {  
            System.err.println("Error");  
        }  
        System.out.println(  
            "myStr: " + myStr + ", myNum: " + myNum);  
    }  
  
    public static void main(String[] args) {  
        App obj = new App();  
        obj.doStuff("9009");  
    }  
}
```

What is the result?

- A. myStr: 9009, myNum: 9009
- B. myStr: 7007, myNum: 7007
- C. myStr: 7007, myNum: 9009
- D. Compilation fails

Answer: C

125. 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

126. Given the code fragment:

```

public static void main(String[] args) {
    int array[] = {10, 20, 30, 40, 50};
    int x = array.length;
    /* line n1 */
}

```

Which two code fragments can be independently inserted at line n1 to enable the code to print the elements of the array in reverse order? (Choose two.)

```

A
while (x > 0) {
    x--;
    System.out.print(array[x]);
}

B
do {
    x--;
    System.out.print(array[x]);
} while (x >= 0);

C
while (x >= 0) {
    System.out.print(array[x]);
    x--;
}

D
do {
    System.out.print(array[x]);
    --x;
} while (x >= 0);

E
while (x > 0) {
    System.out.print(array[--x]);
}

```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

Answer: A E

127. This grid shows the state of a 2D array:

0	0	
	x	0
x		x

The grid is created with this code:

```

char[][] grid = new char[3][3];
grid[1][1] = 'X';
grid[0][0] = 'O';
grid[2][0] = 'X';
grid[0][1] = 'O';
grid[2][2] = 'X';
grid[1][2] = 'O';
//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 Xs?

- A. grid[2][1] = 'X';
- B. grid[3][2] = 'X';
- C. grid[3][1] = 'X';
- D. grid[2][3] = 'X';

Answer: D

128. Given:

```

public class MyField {
    int x;
    int y;
    public void doStuff(int x, int y) {
        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 : 0 0 :

- B. 100 200 : 100 0 :
- C. 100 200 : 100 200 :
- D. 0 0 : 100 0 :

Answer: B

129. Given the code fragment:

```
public static void main(String[] args) {  
    try {  
        int num = 10;  
        int div = 0;  
        int ans = num / div;  
    } catch (ArithmaticException ae) {  
        ans = 0 // line n1  
    } catch (Exception e) {  
        System.out.println("Invalid calculation");  
    }  
    System.out.println("Answer = " + ans); // line n2  
}
```

What is the result?

Answer: = 0

- B. Invalid calculation
- C. Compilation fails only at line n1.
- D. Compilation fails only at line n2.
- E. Compilation fails at line n1 and line2.

Answer: E 130.

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

131. 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. 0:20

C. Compilation fails at line n1

D. Compilation fails at line n2

Answer: D

132. 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 : a) {  
            System.out.print(i+ " ");  
        }  
        System.out.println();  
    }  
}
```

What is the result?

A Compilation fails.

B

```
1 3  
1 3
```

C

```
1 3  
followed by an ArrayIndexOutOfBoundsException
```

D

```
1 3  
1 3 0 0
```

E

```
1 3 5 7  
1 3
```

A. Option A

B. Option B

C. Option C

D. Option D E. Option E

Answer: E

Explanation:

```

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

```

External Libraries ... [Add External Library \(from Maven Repo\)](#)

CommandLine Arguments ...

Interactive mode : OFF

Version: JDK 9.0.1

Stdin Inputs...

Result...

CPU Time: 0.13 sec(s), Memory: 30680 kilobyte(s)

compiled and executed in 0.705 sec(s)

```

1 3 5 7
1 3

```

133. Examine the content of App.java:

```

package p1;
public class App {
    public static void main(String[] args) {
        System.out.println("Java");
    }
}

```

and of Test.java:

```

package p1.p2;
public class Test {}

```

Which is true?

- A. The App.class file is stored within the p1 folder. The Test.class file is stored within the p2 sub-folder of p1.
- B. The App class is accessible within the Test class without an import statement.
- C. import p1.App; is used to access the App class within the Test class.
- D. It is optional to have the package statement as the first line of class definitions.

Answer: C

134. Which two class definitions fail to compile? (Choose two.)

A

```
abstract class A3 {  
    private static int i;  
    public void doStuff() {}  
    public A3() {}  
}
```

B

```
final class A1 {  
    public A1() {}  
}
```

C

```
private class A2 {  
    private static int i;  
    private A2() {}  
}
```

D

```
class A4 {  
    protected static final int i = 10;  
    private A4() {}  
}
```

E

```
final abstract class A5 {  
    protected static int i;  
    void doStuff() {}  
    abstract void doIt();  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D E. Option E

Answer: C D

135. Given this code for a Planet object:

```
public class Planet {  
    public String name;  
    public int moons;  
  
    public Planet(String name, int moons) {  
        this.name = name;  
        this.moons = moons;  
    }  
}
```

And this method:

```
public static void main(String[] args) {  
    Planet[] planets = {  
        new Planet("Mercury", 0),  
        new Planet("Venus", 0),  
        new Planet("Earth", 1),  
        new Planet("Mars", 2)  
    };  
  
    System.out.println(planets);  
    System.out.println(planets[2].name);  
    System.out.println(planets[2].moons);  
}
```

What is the output?

A

```
planets
Earth
1
```

B

```
[LPlanets.Planet;@15db9742
Earth
1
```

C

```
[LPlanets.Planet;@15db9742
Planets.Planet@6d06d69c
1
```

D

```
[LPlanets.Planet;@15db9742
Planets.Planet@6d06d69c
[LPlanets.Moon;@7852e922
```

E

```
[LPlanets.Planet;@15db9742
Venus
0
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D E. Option E

Answer: C

136. Given the code fragment:

```
List<String> arrayList = new ArrayList<>();
arrayList.add("Tech");
arrayList.add("Expert");
arrayList.set(0, "Java");
arrayList.forEach (a -> a.concat("Forum"));
arrayList.replaceAll (s -> s.concat("Group"));
System.out.println(arrayList);
```

What is the result?

- A. [JavaForum, ExpertForum]

- B. [JavaGroup, ExpertGroup]
- C. [JavaForumGroup, ExpertForumGroup]
- D. [JavaGroup, TechGroup ExpertGroup]

Answer: B

Explanation:

```
21+ public class Main {  
22+     public static void main(String[] args) {  
23         List<String> arrayList = new ArrayList<>();  
24         arrayList.add("Tech");  
25         arrayList.add("Expert");  
26         arrayList.set(0, "Java");  
27         arrayList.forEach (a -> a.concat ("Forum"));  
28         arrayList.replaceAll (s -> s.concat("Group"));  
29         System.out.println(arrayList);  
30     }  
31  
32  
33  
34  
35 }
```

CPU Time: 0.18 sec(s), Memory: 32824 kilobyte(s)

[JavaGroup, ExpertGroup]

137. Given the code fragment:

```
int n[][] = {{1, 3}, {2, 4}};  
for (int i = n.length - 1; i >= 0; i--) {  
    for (int j = n[i].length - 1; j >= 0; j--) {  
        System.out.print(n[i][j]);  
    }  
}
```

What is the result?

- A.-3142
- B.2413
- C.1324
- D.4231

Answer: D

Explanation:

The screenshot shows a Java code editor with the file 'Main.java' open. The code defines three classes: C, B, and A. Class C prints "C ". Class B extends C and prints "B ". Class A extends B and prints "A ". The main method creates an instance of A and prints "A ". The code is annotated with line numbers from 1 to 21.

```
1
2 class C {
3     public C() {
4         System.out.print("C ");
5     }
6 }
7
8 class B extends C{
9     public B() {
10        System.out.print("B ");
11    }
12 }
13 public class A extends B{
14
15     public A(){
16         System.out.print("A ");
17     }
18     public static void main(String[] args) {
19         A a = new A();
20     }
21 }
```

The terminal window to the right shows the compilation command and its output. It uses the Java compiler (javac) with a classpath that includes junit-4.12.jar, hamcrest-core-1.3.jar, and json-simple-1.1.1.jar. The output indicates an error: 'Main.java:13: error: class A is public, should be declared in a file named A.java'. The compilation exits with a status of 1.

```
java version "1.8.0_31"
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-core-1.3.jar:/run_dir/json-simple-1.1.1.jar -d . Main.java
Main.java:13: error: class A is public, should be declared in a
file named A.java
public class A extends B{
 ^
1 error
compiler exit status 1
```

138. Which statement is true about Java byte code? A. It can run on any platform.
- B. It can run on any platform only if it was compiled for that platform.
- C. It can run on any platform that has the Java Runtime Environment.
- D. It can run on any platform that has a Java compiler.
- E. It can run on any platform only if that platform has both the Java Runtime Environment and a Java compiler.

Answer: D

Explanation:

Java bytecodes help make "write once, run anywhere" possible. You can compile your program into bytecodes on any platform that has a Java compiler. The bytecodes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

139. Given:

```

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

```

What is the result?

- A. Welcome Visit Count:0Welcome Visit Count: 1
- B. Compilation fails at line n2.
- C. Compilation fails at line n1.
- D. Welcome Visit Count:0Welcome Visit Count: 0

Answer: C

Explanation:

```

1
2 public class App {
3     int count;
4     public static void displayMsg() {
5         System.out.println("Welcome Visit Count: " + count ++); //line n1
6     }
7     public static void main(String[] args) {
8         App.displayMsg();
9         displayMsg();
10    }
11 }
12

```

140. 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.     fVar = iVar;
8.     iVar = fVar;
9.     fVar = dVar;
10.    dVar = fVar;
11.    iVar = dVar;
12.    dVar = iVar;
13. }

```

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

- A. Line 7
- B. Line 8
- C. Line 9
- D. Line 10

E. Line 11

F. Line 12

Answer: A D F

141. 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);  
            System.out.println(pwd[idx]);  
            idx++;  
        }  
    }  
    catch (Exception e) {  
        System.out.println("Invalid Name");  
    }  
}
```

What is the result?

A

omas
Invalid Name
null

B

omas
ter
seph

C Invalid Name

D

omas
Invalid Name

A. Option A

B. Option B

C. Option C

D. Option D

Answer: D

Explanation:

Result

CPU Time: 0.15 sec(s), Memory: 29904 kilobyte(s)

```
omas  
Invalid Name
```

142. 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;  
    }  
}  
System.out.println(wd);
```

What is the result?

- A. 3
- B. 0
- C. Compilation fails.
- D. -1

Answer: D

143. Given:

```
public class Fieldinit {  
    char c;  
    boolean b;  
    float f;  
    void printAll() {  
        System.out.println ("c = " + c);  
        System.out.println ("b = " + b);  
        System.out.println ("f = " + f);  
    }  
    public static void main (String [] args) {  
        FieldInit f = new FieldInit ();  
        f.printAll ();  
    }  
}
```

What is the result?

A

```
c=  
b = false  
f = 0.0
```

B

```
c= null  
b = true  
f = 0.0
```

C

```
c=0  
b = false  
f = 0.0f
```

D

```
c= null  
b = false  
f = 0.0F
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

144. 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 A

B. Option B

C. Option C

D. Option D

Answer: C

145. 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 main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

What is the result?

- A. double sum is 30.0 float sum is 30.0
- B. float sum is 30.0 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

Answer: B 146. 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. Element 0Element 1
- B. Null element 0Null element 1
- C. NullNull
- D. A NullPointerException is thrown at runtime.

Answer: D

147. Given the code fragment:

```
class Employee {  
    private String name;  
    private int age;  
    private int salary;  
  
    public Employee(String name, int age) {  
        setName(name);  
        setAge(age);  
        setSalary(2000);  
    }  
  
    public Employee(String name, int age, int salary) {  
        this(name, age);  
        setSalary(salary);  
    }  
  
    //getter and setter methods for attributes go here  
  
    public void printDetails() {  
        System.out.println(name + " : " + age + " : " + salary);  
    }  
}
```

Test.java:

```
class Test {  
    public static void main(String[] args) {  
        Employee e1 = new Employee();  
        Employee e2 = new Employee("Jack", 50);  
        Employee e3 = new Employee("Chloe", 40, 5000);  
  
        e1.printDetails();  
        e2.printDetails();  
        e3.printDetails();  
    }  
}
```

Which is the result?

A Compilation fails in the Employee class.

B

```
null : 0 : 0  
Jack : 50 : 0  
Chloe : 40 : 5000
```

C

```
null : 0 : 0  
Jack : 50 : 2000  
Chloe : 40 : 5000
```

D Compilation fails in the Test class.

E Both the Employee class and the Test class fail to compile.

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

Answer: D 148. Given the code

fragment:

```
LocalDateTime dt = LocalDateTime.of(2014, 7, 31, 1, 1);  
dt.plusDays(30);  
dt.plusMonths(1);  
System.out.println(dt.format(DateTimeFormatter.ISO_DATE_TIME));
```

What is the result?

A. An exception is thrown at runtime.

B. 2014-07-31T01:01:00 C. 2014-07-31

D. 2014-09-30T00:00:00

Answer: B

149. 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

150. 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 2
- B. java MyFile 1 2 3 4
- C. java MyFile 1 2 2
- D. java MyFile 2 2

Answer: C

151. Given:

```

class Alpha {
    int ns;
    static int s;
    Alpha (int ns) {
        if (s < ns) {
            s = ns;
            this.ns = ns;
        }
    }
    void doPrint () {
        System.out.println("ns= " + ns + " s = " + s);
    }
}

```

And:

```

public class TestA {
    public static static void main(String[] args) {
        Alpha ref1 = new Alpha (100);
        Alpha ref2 = new Alpha (50);
        Alpha ref3 = new Alpha (125);
        ref1.doPrint();
        ref2.doPrint();
        ref3.doPrint();
    }
}

```

What is the result?

A. ns = 100 s = 125

ns = 0 s = 125 ns = 125 s

= 125 B. ns = 50 s = 50 ns

= 125 s = 125 ns = 0 s =

125

C. ns = 50 s = 125 ns = 125 s = 125

ns = 0 s = 125

D. ns = 50 s = 50 ns = 125 s = 125 ns

= 100 s = 100

Answer: C

152. Given the code fragment:

```

10. public static void main(String[] args {
11.     List<String> lst = Arrays.asList("A", "B", "C", "D");
12.     Iterator<String> itr = lst.iterator();
13.     while(itr.hasNext()) {
14.         String e = itr.next();
15.         if (e == "C") {
16.             break;
17.         }
18.         else {
19.             continue;
20.             System.out.print(e);
21.         }
22.     }
23. }
```

Which action enables it to print AB? A.

Comment lines 18 to 21.

B. Comment line 20. C.

Comment line 19.

D. Comment line 16.

Answer: B

153. Given:

```

public class FieldInit {
    Character c;
    boolean b;
    float f;
    void printAll() {
        System.out.println("c = " + c);
        System.out.println("b = " + b);
        System.out.println("f = " + f);
    }

    public static void main(String[] args) {
        FieldInit f = new FieldInit();
        f.printAll();
    }
}
```

What is the result?

A

```
c =  
b = false  
f = 0.0
```

B

```
c = null  
b = true  
f = 0.0
```

C

```
c = 0  
b = false  
f = 0.0F
```

D

```
c = null  
b = false  
f = 0.0
```

A. Option A

B. Option B

C. Option C

D. Option D

Answer: D

Explanation:

```
c = null  
b = false  
f = 0.0
```

```
Completed with exit code: 0
```

154. Given the code fragment:

```

public static void main(String[] args) {
    ArrayList myList = new ArrayList();
    String[] myArray;
    try {
        while (true) {
            myList.add("My String");
        }
    } catch (RuntimeException re) {
        System.out.println("Caught a RuntimeException");
    } catch (Exception e) {
        System.out.println("Caught an Exception");
    }
    System.out.println("Ready to use");
}

```

What is the result?

- A. Execution terminates in the first catch statement, and Caught a RuntimeException is printed to the console.
- B. Execution terminates in the second catch statement, and Caught an Exception is printed to the console.
- C. A runtime error is thrown in the thread "main".
- D. Execution completes normally, and Ready to use is printed to the console.
- E. The code fails to compile because a throws keyword is required.

Answer: C

155. 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++;
        case "wed":
            wd += 2;
    }
}
System.out.println(wd);

```

What is the result?

- A..... 2
- B..... 10

C. -1

D. Compilation fails.

Answer: A

156. Which statement best describes encapsulation?

A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.

B. Encapsulation ensures that classes can be designed so that their methods are inheritable.

C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.

D. Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

Answer: A

157. Given the code fragment:

```
int num[][] = new int[3][1];
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 B

C. Option C

D. Option D

Answer: B

158. Given the code fragment:

```
if (aVar++ < 10) {  
    System.out.println(aVar + " Hello Universe!");  
} else {  
    System.out.println(aVar + " Hello World!");  
}
```

What is the result if the integer aVar is 9? A.

Compilation fails.

B. 10 Hello Universe!

C. 10 Hello World!

D. 9 Hello World!

Answer: B

159. Given:

```
class Book {int pages;}

public class App{
    int count;

    public void method(Book x, int k) {
        x.pages = 100;
        k = 200;
    }

    public static void main(String[] args) {
        App obj = new App();
        Book objBook = new Book();
        System.out.println(objBook.pages + ":" + obj.count);
        obj.method(objBook, obj.count);
        System.out.println(objBook.pages + ":" + obj.count);
    }
}
```

What is the result?

- A. 0:0 100:0
- B. null:0 100:0
- C. 0:0 100:200
- D. null:null 100:null

Answer: A

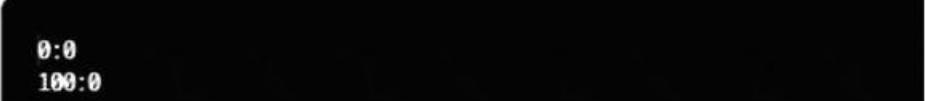
Explanation:

```

15
16  class Book {int pages;}
17  public class App{
18      int count;
19
20      public void method(Book x, int k) {
21          x.pages = 100;
22          k = 200;
23      }
24
25  public static void main(String[] args) {
26      App obj = new App();
27      Book objBook = new Book();
28      System.out.println(objBook.pages + ":" + obj.count);
29      obj.method(objBook, obj.count);
30      System.out.println(objBook.pages + ":" + obj.count);
31  }
32 }
```

Result

CPU Time: 0.24 sec(s), Memory: 35920 kilobyte(s)



```

0:0
100:0
```

160. Given the code fragment:

```

public class Test {
    public static void main(String[] args) {
        int x;
        /* insert code here */
    }
}
```

Which two code fragments inserted at line 10 print ****?

A

```
x = 3;
do {
    System.out.print("*");
    x--;
} while (x >= 0);
```

B

```
x = 0;
do {
    System.out.print("*");
    x++;
} while (x >= 3);
```

C

```
x = 0;
do {
    System.out.print("*");
    ++x;
}while (x > 3);
```

D

```
x = 3;
do {
    System.out.print("*");
    x--;
}while (x != 1);
```

E

```
x = 0;
do {
    System.out.print("*");
} while (x++ < 3);
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

Answer: E

161. Which two initialization statements are valid? (Choose two.) A.

Boolean available = "TRUE":

B. String tmpAuthor = author, author ="Mc Donald";

C. Double price = 200D;

D. Integer pages = 20;

Answer: C D

162. Given the code fragments:

```
class Student {  
    String name;  
    int age;  
}
```

And:

```
4. public class Test {  
5.     public static void main(String[] args) {  
6.         Student s1 = new Student();  
7.         Student s2 = new Student();  
8.         Student s3 = new Student();  
9.         s1 = s3;  
10.        s3 = s2;  
11.        s2 = null;  
12.    }  
13.}
```

Which statement is true?

- A. After line 11, three objects are eligible for garbage collection.
- B. After line 11, two objects are eligible for garbage collection.
- C. After line 11, one object is eligible for garbage collection.
- D. After line 11, none of the objects are eligible for garbage collection.

Answer: C

163. Given:

```
public class Test {  
    public static void main(String[] args) {  
        int x = 1;  
        int y = 0;  
        if(x++ > ++y) {  
            System.out.print("Hello ");  
        } else {  
            System.out.print("Welcome ");  
        }  
        System.out.print("Log " + x + ":" + y);  
    }  
}
```

What is the result?

- A. Hello Log 1:0

B. Hello Log 2:1

C. Welcome Log 2:1

D. Welcome Log 1:0

Answer: C

164. Given the following code:

```
int[] intArr = {15, 30, 45, 60, 75};  
intArr[2] = intArr[4];  
intArr[4] = 90;
```

What are the values of each element in intArr after this code has executed?

A. 15, 60, 45, 90, 75

B. 15, 90, 45, 90, 75

C. 15, 30, 75, 60, 90

D. 15, 30, 90, 60, 90

E. 15, 4, 45, 60, 90

Answer: C

165. Given:

```
public class OraString {  
    String s;  
    public boolean equals(OraString str) {  
        return this.s.equalsIgnoreCase(str.toString());  
    }  
    OraString(String s) {  
        this.s = s;  
    }  
}
```

and the code fragment:

```
String s1 = "Moon";
OraString s2 = new OraString("Moon");

if ((s1 == "Moon") && (s2.equals("Moon"))) {
    System.out.print("A");
} else {
    System.out.print("B");
}
if (s1.equalsIgnoreCase(s2.s)) {
    System.out.print("C");
} else {
    System.out.print("D");
}
```

What is the result?

- A. AC
- B. BD
- C. BC
- D. AD

Answer: C

Explanation:

```

-- 16- public class OraString {
17     String s;
18-     public boolean equals(OraString str) {
19         return this.s.equalsIgnoreCase(str.toString());
20     }
21- }
22- OraString(String s) {
23     this.s = s;
24 }
25
26- public static void main(String[] args) {
27     String s1 = "Moon";
28     OraString s2 = new OraString("Moon");
29
30-     if ((s1 == "Moon") && (s2.equals("Moon"))) {
31         System.out.println("A");
32     } else {
33         System.out.println("B");
34     }
35-     if (s1.equalsIgnoreCase(s2.s)) {
36         System.out.println("C");
37     } else {
38         System.out.println("D");
39     }
40 }
41 }
--
```

Result

CPU Time: 0.16 sec(s), Memory: 32160 kilobyte(s)

B
C

166. Which two statements are true about Java byte code? (Choose two.) A. It can be serialized across network.
 B. It can run on any platform that has a Java compiler.
 C. It can run on any platform.
 D. It has “java” extension.
 E. It can run on any platform that has the Java Runtime Environment.

Answer: A E

167. Given the code fragment:

```

int x = 100;
int a = x++;
int b = ++x;
int c = x++;
int d = (a < b) ? (a < c) ? a: (b < c) ? b: c: x;
System.out.println(d);
```

What is the result?

- A. 100
- B. 101
- C. 102
- D. 103

E.Compilation fails

Answer: E

168. 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

B

Invalid Name
omas

C

Invalid Name
omas
null
null

D

omas
ter
seph

A. Option A

B. Option B

C. Option C

D. Option D

Answer: C

169. Given:

```
package clothing;
public class Shirt {
    public static 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 insert: import clothing.Shirt;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 static clothing.Shirt.getColor;At line n2 insert: String color = getColor();
- D. At line n1 no changes required.At line n2 insert: String color = Shirt.getColor();
- E. At line n1 insert: import Shirt;At line n2 insert: String color = Shirt.getColor();

Answer: A

170. 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. employee.salary = 50_000;
- B. director.salary = 80_000;
- C. employee.budget = 200_000;
- D. manager.budget = 1_000_000;
- E. manager.stockOption = 500;
- F. director.stockOptions = 1_000;

Answer: C E

171. Which three statements are true about exception handling? (Choose three.) A. Only unchecked exceptions can be rethrown.

- B. All subclasses of the RuntimeException class are not recoverable.
- C. The parameter in a catch block is of Throwable type.
- D. All subclasses of the RuntimeException class must be caught or declared to be thrown.
- E. All subclasses of the RuntimeException class are unchecked exceptions.
- F. All subclasses of the Error class are not recoverable.

Answer: B C D

172. 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. 5 4 3 2 1 0
- B. 5 4 3 2 1
- C. 4 2 1
- D. 5
- E. Nothing is printed

Answer: D

173. Given these requirements:

- » Bus and Boat are Vehicle type classes.
- » The start() and stop() methods perform common operations across the Vehicle class type.
- » The ride() method performs a unique operations for each type of Vehicle.

Which set of actions meets the requirements with optimized code?

A. * 1. Create an abstract class Vehicle by defining start() and stop() methods, and declaring the ride() abstract method.

* 2. Create Bus and Boat classes by inheriting the Vehicle class and overriding the ride() method.

B. * 1. Create an interface Vehicle by defining start() and stop() methods, and declaring the ride() abstract method.

* 2. Create Bus and Boat classes by implementing the Vehicle class.

C. * 1. Create an abstract class Vehicle by declaring stop(), start(), and ride() abstract methods.

* 2. Create Bus and Boat classes by inheriting the Vehicle class and overriding all the methods. D. * 1.

Create an interface Vehicle by defining default stop(), start(), and ride() methods.

* 2. Create Bus and Boat classes by implementing the Vehicle interface and overriding the ride() method.

Answer: B

174. Given the definitions of the MyString class and the Test class:

```
package p1;
class MyString {
    String msg;
    MyString(String msg) {
        this.msg = msg;
    }
}
```

Test.java:

```
package p1;
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello " + new StringBuilder("Java SE 8"));
        System.out.println("Hello " + new MyString("Java SE 8").msg);
    }
}
```

What is the result?

A
Hello Java SE 8
Hello Java SE 8

B
Hello java.lang.StringBuilder@<<hashcode1>>
Hello p1.MyString@<<hashcode2>>

C
Hello Java SE 8
Hello p1.MyString@<<hashcode>>

D Compilation fails at the Test class

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

Answer: D

175. Given:

```
public class App {  
    int foo;  
    static int bar;  
  
    static void process() {  
        foo += 10;  
        bar += 10;  
    }  
    public static void main(String[] args) {  
        App firstObj = new App();  
        App.process();  
        System.out.println(firstObj.bar);  
  
        App secondObj = new App();  
        App.process();  
        System.out.println(secondObj.bar);  
    }  
}
```

What is the result?

- A. 10
- * 20
- B. A compile time error occurs

C. 20

*20

D. 10 *10

Answer: B

Explanation:

Result
CPU Time: sec(s), Memory: kilobyte(s)

```
/App.java:21: error: non-static variable foo cannot be referenced from a static context
    foo +=10;
    ^
1 error
```

176. Given the code fragment:

```
abstract class Robot implements Speakable {
    public void process();
}

class Humanoid extends Robot {
    public void speak(String s) { System.out.println(s); }
    public void process() { System.out.println("Helping... "); }
}

interface Speakable {
    public void speak(String s);
}

public class RobotApp{
    public static void main(String[] args) {
        Robot r = new Humanoid();
        r.process();
        r.speak("Done");
    }
}
```

Which action enables the code to print Helping... Done? A.

replace class Humanoid extends Robot { with abstract class

Humanoid extends Robot {

B. replace interface Speakable { with abstract class Speakable C.

replace public void process(); with public abstract void process();

D. replace abstract class Robot implements Speakable { with class Robot extends Speakable {

Answer: C

177. Given the definitions of the Bird class and the Peacock class:

```
public class Bird {  
    public void fly() {  
        System.out.print ("Fly.");  
    }  
}  
  
public class Peacock extends Bird {  
    public void dance() {  
        System.out.print("Dance.");  
    }  
}
```

and the code fragment:

```
/*insert code snippet here */  
p.fly();  
p.dance();
```

Which code snippet can be inserted to print Fly.Dance. ?

- A. Bird p = new Peacock();
- B. Bird b = new Bird();Peacock p = (Peacock) b;
- C. Peacock b = new Peacock ();Bird p = (Bird) b;
- D. Bird b = new Peacock ();Peacock p = (Peacock) b;

Answer: B

178. Which is true about the switch statement?

- A. Its expression can evaluate to a collection of values.
- B. The break statement, at the end of each case block, is optional.
- C. Its case label literals can be changed at runtime.
- D. It must contain the default section.

Answer: B

179. 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

180. 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 B

C. Option C

D. Option D

Answer: B C

181. Given the content of three files:

A.java:

```
public class A {  
    public void a() {}  
    int a;  
}
```

B.java:

```
public class B {  
    private int doStuff() {  
        private int x = 100;  
        return x++;  
    }  
}
```

C.java:

```
import java.io.*;  
package pl;  
class A {  
    public void main(String fileName) throws IOException {}  
}
```

Which statement is true?

A. Only the A.Java file compiles successfully.

- B. Only the B.java file compiles successfully.
- C. Only the C.java file compiles successfully.
- D. The A.java and B.java files compile successfully.
- E. The B.java and C.java files compile successfully.
- F. The A.java and C.java files compile successfully.

Answer: A

182. Examine the given definitions:

```
class Player {}

interface Playable {
    public void play();
    public void setPlayers(List<Player> players);
}

class Game implements Playable {
    private List<Player> players;
    public List<Player> getPlayers() { return players; }
    public void setPlayers(List<Player> players) { this.players
= players; }
    public void play() { System.out.println("Played."); }
}
```

and the code fragment:

```
Playable p = new Game();
List<Player> players = new ArrayList<>();
p.setPlayers (players);
p.play();
```

Which statement is true about the implementation of Object-Oriented Programming concepts in the given code?

- A. Polymorphism, abstraction, and encapsulation are implemented.
- B. Only polymorphism and inheritance are implemented.
- C. Polymorphism, inheritance, and abstraction are implemented.
- D. Only inheritance and encapsulation are implemented.

Answer: C

183. 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: D

184. Given this 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 * height whenever the Rectangle class is used?

A. Call the setArea method at the end of the setHeight method.

B. Call the setArea method at the beginning of the setHeight method.

C. Call the setArea method at the end of the setLength method.

D. Call the setArea method at the beginning of the setLength method.

E. Change the setArea method to private.

F. Change the area field to public.

Answer: A E

185. Given the code fragments:

A.java:

```
package p1;
public class A {
```

B.java:

```
package p1.p2;
//line n1
public class B {
    public void doStuff() {
        A b = new A();
    }
}
```

C.java:

```
package p3;
//line n2
public class C {
    public static void main(String[] args) {
        A o1 = new A();
        B o2 = new B();
    }
}
```

Which modification enables the code to compile?

A

Replace line n1 with:
import p1.*;
Replace line n2 with:
import p1. p2.*;

B

Replace line n1 with:
import p1. A;
Replace line n2 with:
import p1.*;

C

Replace line n1 with:
import p1. A;
Replace line n2 with:
import p1. A;
import p1. p2.B ;

D

Replace line n1 with:
import p1;

Replace line n2 with:
import p1;
import p1.p2;

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: C

186. Given:

```
class S1 {  
    protected void display(int x) {  
        System.out.print("Parent" + x);  
    }  
}  
class S2 extends S1 {  
    public void display(int x, int y) {  
        this.display(x);  
        display(y);  
        super.display(y);  
    }  
    public void display(int x) {  
        System.out.println("Child " + x);  
    }  
}
```

and the code fragment:

S2 sobj = new S2(); sobj.display(10, 100); What is the result?

A. Child 10 Child 100

Parent 100

B. Parent 10 Child 10

Parent 1000

C. Child 10

Parent 100

Parent 100

D. A compile time error occurs.

Answer: D

Explanation:

```
|  
| Error: Main method not found in class S1, please define the main method as:  
|   public static void main(String[] args)  
| or a JavaFX application class must extend javafx.application.Application  
|
```

187. Given:

```

public class Test {

    public static void main(String[] args) {
        if (args[0].equals("Hello") ? true : false) {
            System.out.println("Success");
        } else {
            System.out.println("Failure");
        }
    }
}

```

And given the commands:

```

javac Test.java
Java Test Hello

```

What is the result?

- A. Success
- B. Failure
- C. Compilation fails.
- D. An exception is thrown at runtime

Answer: A

188. Given this class:

```

public class CheckingAccount {
    public int amount;
    public CheckingAccount(int amount) {
        this.amount = amount;
    }
    public int getAmount(){ return amount; }
    public void setAmount(int amount){ this.amount = amount; }
    public void changeAmount(int x) {
        amount += x;
    }
}

```

And given this main method, located in another class:

```

public static void main(String[] args) {
    CheckingAccount acct = new CheckingAccount((int)(Math.random()*1000));
    //line n1
    System.out.println(acct.getAmount());
}

```

Which three lines, when inserted independently at line n1, cause the program to print a 0 balance?

- A. acct.setAmount(-acct.getAmount());
- B. acct.amount = 0; <option D earlier>

- C. acct.setAmount(0);
- D. acct.getAmount() = 0; <option E earlier>
- E. this.amount = 0; <option A earlier>
- F. acct.changeAmount(0); <option F earlier>
- G. acct.changeAmount(-acct.amount); <option G earlier>

Answer: B D F

189. Given:

```
public class Test {  
    public static int stVar = 100;  
    public int var = 200;  
    public String toString() {  
        return stVar + ":" + var;  
    }  
}
```

And given the code fragment:

```
Test t1 = new Test();  
t1.var = 300;  
System.out.println(t1);  
Test t2 = new Test();  
t2.stVar = 300;  
System.out.println(t2);
```

What is the result?

- A. 300:300200:300
- B. 300:100200:300
- C. 300:00:300
- D. 100:300300:200

Answer: D

190. Examine:

```
class E1 extends Exception { }

class E2 extends RuntimeException { }

public class App {
    public void m1() {
        System.out.println("m1.Accessed.");
        throw new E1();
    }

    public void m2 () {
        System.out.println("m2.Accessed.");
        throw new E2();
    }
    public static void main (String[] args) {
        int level =1;
        App obj = new App();
        if (level <=5 && level >= 3) {
            obj.m1();
        } else {
            obj.m2 ();
        }
    }
}
```

Which statement is true?

- A. The program prints m1.Accessed.
- B. The program fails compile due to the unhandled E1 exception.
- C. The program prints m2.Accessed.
- D. The program fails to compile due to the unhandled E2 exception.

Answer: B

Explanation:

```
4 public class App {
5     public void m1() {
6         System.out.println("m1.Accessed.");
7         throw new E1();
8     }
9
10    public void m2 () {
11        System.out.println("m2.Accessed.");
12        throw new E2 ();
13    }
14
15    public static void main (String[] args) {
16        int level =1;
17        App obj = new App();
18        if (level <=5 && level >= 3) {
19            obj.m1();
20
21        } else {
22            obj.m2();
23        }
24    }
25 }
26 }
```

191. 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 = 'o';
        obj2.var = 'i';

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

What is the result?

- A. a, ei, i
- B. a, eo, o
- C. e, ei, i
- D. a, ao, o

Answer: A

192. 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. Requirements 1, 2, and 3 can be implemented by using the enhanced for loop.
- B. Requirements 1, 2, and 3 can be implemented by using the standard for loop.
- C. Requirements 2 and 3 CANNOT be implemented by using the standard for loop.
- D. Requirement 1 can be implemented by using the enhanced for loop.
- E. Requirement 3 CANNOT be implemented by using either the enhanced for loop or the standard for loop.

Answer: D E

193. Given the code fragment:

```
public static void main(String[] args) {  
    int sum = 0;  
    for(int xVal = 1; xVal <= 5; xVal++) {  
        sum = sum + xVal;  
    }  
    System.out.print("The sum of " + xVal + " numbers is: " + sum);  
}
```

What is the result?

- A. The sum of 4 numbers is: 10
- B. A compile time error occurs.
- C. The sum of 5 numbers is: 10
- D. The sum of 5 numbers is: 15

Answer: B

Explanation:

```
/Main.java:29: error: cannot find symbol
    System.out.print("The sum of " + xVal + "numbers is:" + sum);
                           ^
  symbol:   variable xVal
  location: class Main
1 error
```

194. Given the code fragment:

```
1. public class Test {
2.     public static void main(String[] args) {
3.         /* insert code here */
4.         array[0]=10;
5.         array[1]=20;
6.         System.out.print(array[0]+":"+array[1]);
7.     }
8. }
```

Which code fragment, when inserted at line 3, enables the code to print 10:20?

- A
int[] array = new int[1];
- B
int[] array;
array = new int[2];
- C
int array = new int[2];
- D
int array[1];

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: B

Explanation:

Your Code ...

```
1 ~ public class Test {  
2 ~     public static void main (String[] args) {  
3 ~         int[] array;  
4 ~         array = new int[2];  
5 ~         array[0]=10;  
6 ~         array[1]=20;  
7 ~         System.out.print(array[0]+ ":" +array[1]);  
8 ~     }  
9 ~ }  
10
```

CommandLine Arguments ...

Stdin Inputs...

Execute

Save

My I

Result...

CPU Time: 0.10 sec(s), Memory: 30316 kilobyte(s)

10:20

195. Given the code fragment:

```
String shirts[][] = new String[2][2];  
shirts[0][0] = "red";  
shirts[0][1] = "blue";  
shirts[1][0] = "small";  
shirts[1][1] = "medium";
```

Which code fragment prints red: blue: small: medium?

```

A
for (int index = 1; index < 2; index++) {
    for (int idx = 1; idx < 2; idx++) {
        System.out.print(shirts[index][idx] + ":" );
    }
}

B
for (int index = 0; index < 2; ++index) {
    for (int idx = 0; idx < index; ++idx) {
        System.out.print(shirts[index][idx] + ":" );
    }
}

C
for (String [] c : shirts) {
    for (String s : c) {
        System.out.print(s + ":" );
    }
}

D
for (int index = 0; index <=2;) {
    for (int idx = 0; idx <=2;) {
        System.out.print(shirts[index][idx] + ":" );
        idx++;
    }
    index++;
}

```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: D

196. Given this segment of code:

```
ArrayList<Cycle> myList = new ArrayList<>();
myList.add(new MotorCycle());
```

Which two statements, if either were true, would make the code compile? (Choose two.) A.

- MotorCycle is an interface that implements the Cycle class.
- B. Cycle is an interface that is implemented by the MotorCycle class.
- C. Cycle is an abstract superclass of MotorCycle.
- D. Cycle and MotorCycle both extend the Transportation superclass.
- E. Cycle and MotorCycle both implement the Transportation interface.

F. MotorCycle is a superclass of Cycle.

Answer: B C

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

public class must have a main method.

B. A class can have only one private constructors.

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

D. A class can have overloaded static methods.

E. The methods are mandatory components of a class.

F. The fields need not be initialized before use.

Answer: A C E

198. 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 A

B. Option B

C. Option C

D. Option D

E. Option E

Answer: E

199. Given:

```
String stuff = "TV";
String res = null;

if (stuff.equals("TV")) {
    res = "Walter";
} else if (stuff.equals("Movie")) {
    res = "White";
} else {
    res = "No Result";
}
```

Which code fragment can replace the if block?

A
stuff.equals ("TV") ? res= "Walter" : stuff.equals ("Movie") ?
res = "White" : res = "No Result";

B
res = stuff.equals ("TV") ? "Walter" else stuff.equals
("Movie")? "White" : "No Result";

C
res = stuff.equals ("TV") ? stuff.equals ("Movie")? "Walter" :
"White" : "No Result";

D
res = stuff.equals ("TV")? "Walter" : stuff.equals ("Movie")?
"White" : "No Result";

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: D

200. Given:

```
class Patient {  
    String name;  
    public Patient (String name) {  
        this.name = name;  
    }  
}
```

And the code fragment:

```
8. public class Test {  
9.     public static void main (String [] args) {  
10.         List ps = new ArrayList ();  
11.         Patient p2 = new Patient ("Mike");  
12.         ps.add(p2);  
13.  
14.         // insert code here  
15.  
16.         if (f >= 0) {  
17.             System.out.print ("Mike Found");  
18.         }  
19.     }  
20. }
```

Which code fragment, when inserted at line 14, enables the code to print Mike Found?

A

```
int f = ps.indexOf (p2);
```

B

```
int f = ps.indexOf (Patient ("Mike") );
```

C

```
int f = ps.indexOf (new Patient "Mike") );
```

D

```
Patient p = new Patient("Mike");  
int f = ps.indexOf(p)
```

A. Option A

B. Option B

C. Option C

D. Option D

Answer: A

201. Given:

```

class Student {
    String name;
    public Student(String name) {
        this.name = name;
    }
}

public class Test {
    public static void main(String[] args) {
        Student[] students = new Student[3];
        students[1] = new Student("Richard");
        students[2] = new Student("Donald");
        for (Student s : students) {
            System.out.println(" " + s.name);
        }
    }
}

```

What is the result?

- A. nullRichardDonald
- B. RichardDonald
- C. Compilation fails.
- D. An `ArrayIndexOutOfBoundsException` is thrown at runtime.
- E. A `NullPointerException` is thrown at runtime.

Answer: E 202. Given these

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 this main method:

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

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

- A. director.stockOptions = 1_000;
- B. employee.salary = 50_000;
- C. manager.budget = 1_000_000;
- D. manager.stockOption = 500;
- E. employee.budget = 200_000;
- F. director.salary = 80_000;

Answer: B F

203. 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. 5
- D. 10

Answer: A

204. Which three statements describe the object-oriented features of the Java language? (Choose three.) A.

- Objects cannot be reused.
- B. A subclass must override the methods from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain a main class.
- E. Object is the root class of all other objects.
- F. A main method must be declared in every class.

Answer: B C F

205. 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
- B. 2012-01-30
- C. 2012-02-10 00:00
- D. A DateTimeException is thrown at runtime.

Answer: C

206. Given the code fragment:

```
int a[] = {1, 2, 3, 4, 5};  
for(XXX) {  
    System.out.print(a[e]);  
}
```

Which option can replace xxx to enable the code to print 135?

A

```
int e = 0; e <= 4; e++
```

B

```
int e = 0; e < 5; e += 2
```

C

```
int e = 1; e <= 5; e += 1
```

D

```
int e = 1; e < 5; e += 2
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: B 207. 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 3 1 3
- C. 1 3 1 3 0 0
- D. 1 3 followed by an ArrayIndexOutOfBoundsException
- E. Compilation fails.

Answer: B

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-core-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-core-1.3.jar:/run_dir/json-simple-1.1.1.jar Main  
1 2  
1 3
```

208. Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start(); // line n1  
        c.init(); // line n2  
    }  
}
```

What is the result?

- A. Compilation fails at line n1.
- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails at line n2.

Answer: D 209.

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

Explanation:

```

1 public class Main {
2
3     FileIO Status
4
5     all io completed
6
7
8     public void initialize(int x, int y) {
9         this.x = x * x;
10        this.y = y * y;
11    }
12
13    public static void main(String[] args) {
14        int x = 9, y = 5;
15        Test obj = new Test(x, y);
16        System.out.print(x + " " + y);
17    }
18 }
```

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-core-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-core-1.3.jar:/run_dir/json-simple-1.1.1.jar Main
9 5

210. Given the code fragment:

```

7. public static void main(String[] args) {
8. Predicate<Integer> p = (n) -> n % 2 == 0;
9. // insert code here
10. }
```

Which code snippet at line 9 prints true?

- A. Boolean s = p.apply(101);
System.out.println(s);
- B. Boolean s = p.test(100);
System.out.println(s);
- C. Integer s = p.test(100);
if (s == 1) {
 System.out.println("false");
}
else {
 System.out.println("true");
}
- D. System.out.println(p.apply(100));

A. Option A

B. Option B

C. Option C

D. Option D

Answer: B

211. Which two are benefits of polymorphism? (Choose two.)

A. Faster code at runtime

B. More efficient code at runtime

C. More dynamic code at runtime

D. More flexible and reusable code

E. Code that is protected from extension by other classes

Answer: B D

212. Given:

```
class P1 {}  
class P2 extends P1 implements I1 {}  
interface I1 {}
```

and the code fragment:

```
P1 obj = new P1();  
P2 obj2 = new P2();  
I1 obj3 = new P2();  
boolean r1 = obj instanceof P2;  
boolean r2 = obj2 instanceof P1;  
boolean r3 = obj3 instanceof I1;  
System.out.println(r1 + ":" + r2 + ":"+r3);
```

What is the result?

A. true:false:true

B. false:true:true

C. false:true:false D. true:true:false

Answer: B

213. Given the code fragment:

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

Which three code fragments are valid at line n1?

A

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

B

```
final Toy getToy() {  
    return new Toy();  
}
```

C

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

D

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

E

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

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

Answer: C D E

214. 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 1

```

What is the result?

- A. Java SE
- B. Java EE
- C. Compilation fails at line n1.
- D. A NullPointerException is thrown at runtime.

Answer: B

215. Given:

```

class LogFileNotFoundException extends Exception {}
class AccessViolationException extends RuntimeException {}

1. public class App {
2.     public static void main (String[] args) throws LogFileNotFoundException {
3.         App obj = new App ();
4.         try {
5.             obj.open();
6.             obj.process();
7.             //insert code here
8.         }
9.         catch (Exception e) {
10.             System.out.println("Completed.");
11.         }
12.     }
13.     public void process() {
14.         System.out.println("Processed");
15.         throw new LogFileNotFoundException();
16.     }
17.     public void open () {
18.         System.out.println ("Opened.");
19.         throw new AccessViolationException();
20.     }
21. }

```

Which action fixes the compiler error?

- A. At line 17, add throws AccessViolationException
- B. At line 13, add throws LogFileException
- C. At line 2, replace throws LogFileException with throws AccessViolationException
- D. At line 7, insert throw new LogFileException();

Answer: D

216. 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. 200.0 : 100.0
- B. 400.0 : 200.0 C. 400.0 : 100.0
- D. Compilation fails.

Answer: C

217. Given:

```
public class MyClass {  
    public static void main(String[] args) {  
        String s = "Java SE 8 1";  
        int len = s.trim().length();  
        System.out.print(len);  
    }  
}
```

What is the result?

- A. Compilation fails.
- B. 11
- C. 8
- D. 9
- E. 10

Answer: B

218. Given the code fragment:

```
public class App {  
    public static void main(String[] args) {  
        String str1 = "Java";  
        String str2 = new String("java");  
        //line n1  
        {  
            System.out.println("Equal");  
        } else {  
            System.out.println("Not Equal");  
        }  
    }  
}
```

Which code fragment, when inserted at line n1, enables the App class to print Equal?

- A) str1.toLowerCase();
if (str1 == str2)
- B) if (str2.equals(str1.toLowerCase()))
- C) Str1.toLowerCase();
if (str1.equals(str1.toLowerCase()))
- D) if (str1.toLowerCase() == str2.toLowerCase())

- A. Option A
- B. Option B
- C. Option C

D. Option D

Answer: B

219. 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 main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

What is the result?

A

```
float sum is 30.0  
double sum is 30.0
```

B

```
double sum is 30.0  
float 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 A

B. Option B

C. Option C

D. Option D

Answer: A

220. Given:

```
MainTest.java:  
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 1
- B. An exception is thrown at runtime
- C. String main 1 2 3
- D. String main 123

Answer: A 221. Given the code

fragment:

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

What is the result?

- A. [green, red, yellow, cyan]
- B. [green, blue, yellow, cyan]
- C. [green, red, cyan, yellow]
- D. An IndexOutOfBoundsException is thrown at runtime.

Answer: D 222. Given the code

fragment:

```
List<String> lst = Arrays.asList("EN", "FR", "CH", "JP");
Iterator<String> itr = lst.iterator();
while(itr.hasNext()) {
    String e = itr.next();
    if (e == "CH") {
        break;
    }
System.out.print(e + " ");
}
```

What is the result?

- A. EN FR JP
- B. EN FR
- C. CH
- D. EN FR CH

Answer: B

Explanation:

```
16+ public class Main {
17+     public static void main(String[] args) {
18         List<String> lst = Arrays.asList("EN", "FR", "CH", "JP");
19         Iterator<String> itr = lst.iterator();
20+     while(itr.hasNext()) {
21         String e = itr.next();
22+         if(e == "CH") {
23             break;
24         }
25         System.out.print(e+ " ");
26     }
27 }
28 }
```

Result

CPU Time: 0.28 sec(s), Memory: 35336 kilobyte(s)



EN FR

223. 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 followed by an `ArrayIndexOutOfBoundsException`
- B. 1 2 3
- C. 1 2 3 4
- D. Compilation fails.

Answer: B

Explanation:

```
Console 8 ✘ | Console 9 ✘ | Console 10 ✘  
1 2 3  
Completed with exit code: 0
```