***Operators***

**Q: 01 Given:**

**11. public class Test {**

**12. public static void main(String [] args) {**

**13. int x = 5;**

**14. boolean b1 = true;**

**15. boolean b2 = false;**

**16.**

**17. if ((x == 4) && !b2 ){**

**18. System.out.print("1 ");**

**19. System.out.print("2 ");}**

**20. if ((b2 = true) && b1 )**

**21. System.out.print("3 ");**

**22. }**

**23. }**

**What is the result?**

A. 2

B. 3

C. 1 2

D. 2 3

E. 1 2 3

F. Compilation fails.

G. An exception is thrown at runtime.

**Explanation:**

**Correct Option : B**

It gives the output 3 because first if block which is from 17 to 19 it will give false because (x==4)(here comparing the value) Condition is false so it will not execute its body.

In Line 20 (b2=true)(assigning the value) && operator checks if all the condition true then only executes its block of code.

**Q: 02 Given the command line java Pass2 and:**

**15. public class Pass2 {**

**16. public void main(String [] args) {**

**17. int x = 6;**

**18. Pass2 p = new Pass2();**

**19. p.doStuff(x);**

**20. System.out.print(" main x = " + x);**

**21. }**

**22.**

**23. void doStuff(int x) {**

**24. System.out.print(" doStuff x = " + x++);**

**25. }**

**26. }**

**What is the result?**

A. Compilation fails.

B. An exception is thrown at runtime.

C. doStuff x = 6 main x = 6

D. doStuff x = 6 main x = 7

E. doStuff x = 7 main x = 6

F. doStuff x = 7 main x = 7

**Correct Option : B**

**It will give error at run time because main method is not static**

**Q: 03 Given:**

**13. public class Pass {**

**14. public static void main(String [] args) {**

**15. int x = 5;**

**16. Pass p = new Pass();**

**17. p.doStuff(x);**

**18. System.out.print(" main x = " + x);**

**19. }**

**20.**

**21. void doStuff(int x) {**

**22. System.out.print(" doStuff x = " + x++);**

**23. }**

**24. }**

**What is the result?**

A. Compilation fails.

B. An exception is thrown at runtime.

C. doStuff x = 6 main x = 6

D. doStuff x = 5 main x = 5

E. doStuff x = 5 main x = 6

F. doStuff x = 6 main x = 5

**Correct Option : D**

**The scope of doStuff method variable is only inside the method x will be incremented but the scope of that variable is only limited to the method.**

**Question: 04**

**Given:**

**42. public class ClassA {**

**43. public int getValue() {**

**44.int value=0;**

*45.* **boolean setting = true;**

**46. String title=”Hello”;**

**47. if (value || (setting && title == “Hello”)) { return 1; }**

**48. if (value == 1 & title.equals(”Hello”)) { return 2; }**

**49. }**

*50.* **}**

**And:**

**70. ClassA a = new ClassA();**

**71. a.getValue();**

**What is the result?**

A. 1

B. 2

C. Compilation fails.

D. The code runs with no output.

E. An exception is thrown at runtime.

**Correct Option : C due to line number 47 and return statement is also missing**

**5. Given:**

**class Hexy {**

**public static void main(String[] args) {**

**Integer i = 42;**

**String s = (i<40)?"life":(i>50)?"universe":"everything";**

**System.out.println(s);**

**} }**

**What is the result?**

A. null

B. life

C. universe

D. everything

E. Compilation fails.

F. An exception is thrown at runtime.

**Correct Option : D**

**It works like if else**

**Explanation : If(i<40){System.out.prirntln(“life”);}**

**else if(i>50){ System.out.prirntln(“universe”);}**

**else{ System.out.prirntln(“everything”);}**

**6. Given:**

**1. class Example {**

**2. public static void main(String[] args) {**

**3. Short s = 15;**

**4. Boolean b;**

**5. // insert code here**

**6. }**

**7. }**

**Which, inserted independently at line 5, will compile? (Choose all that apply.)**

A. b = (Number instanceof s);

B. b = (s instanceof Short);

C. b = s.instanceof(Short);

D. b = (s instanceof Number);

E. b = s.instanceof(Object);

F. b = (s instanceof String);

**7. Given:**

**1. class Comp2 {**

**2. public static void main(String[] args) {**

**3. float f1 = 2.3f;**

**4. float[][] f2 = {{42.0f}, {1.7f, 2.3f}, {2.6f, 2.7f}};**

**5. float[] f3 = {2.7f};**

**6. Long x = 42L;**

**7. // insert code here**

**8. System.out.println("true");**

**9. }**

**10. }**

**And the following five code fragments:**

**F1. if(f1 == f2)**

**F2. if(f1 == f2[2][1])**

**F3. if(x == f2[0][0])**

**F4. if(f1 == f2[1,1])**

**F5. if(f3 == f2[2])**

**What is true?**

A. One of them will compile, only one will be true.

B. Two of them will compile, only one will be true.

C. Two of them will compile, two will be true.

D. Three of them will compile, only one will be true.

E. Three of them will compile, exactly two will be true.

F. Three of them will compile, exactly three will be true.

**8. Given:**

**class Fork {**

**public static void main(String[] args) {**

**if(args.length == 1 | args[1].equals("test")) {**

**System.out.println("test case");**

**} else {**

**System.out.println("production " + args[0]);**

**} } }**

**And the command-line invocation:**

**java Fork live2**

**What is the result?**

A. test case

B. production

C. test case live2

D. Compilation fails.

E. An exception is thrown at runtime.

**9. Given:**

**class Foozit {**

**public static void main(String[] args) {**

**Integer x = 0;**

**Integer y = 0;**

**for(Short z = 0; z < 5; z++)**

**if((++x > 2) || (++y > 2))**

**x++;**

**System.out.println(x + " " + y);**

**} }**

**What is the result?**

A. 5 1 B. 5 2

C. 5 3 D. 8 1

E. 8 2 F. 8 3

G. 10 2

H. 10 3

**Correct Option : E**

**10. Given:**

**class Titanic {**

**public static void main(String[] args) {**

**Boolean b1 = true;**

**boolean b2 = false;**

**boolean b3 = true;**

**if((b1 & b2) | (b2 & b3) & b3)**

**System.out.print("alpha ");**

**if((b1 = false) | (b1 & b3) | (b1 | b2))**

**System.out.print("beta ");**

**} }**

**What is the result?**

A. beta

B. alpha

C. alpha beta

D. Compilation fails.

E. No output is produced.

F. An exception is thrown at runtime.

**Correct Option : E**

**11. Given:**

**class Feline {**

**public static void main(String[] args) {**

**Long x = 42L;**

**Long y = 44L;**

**System.out.print(" " + 7 + 2 + " ");**

**System.out.print(foo() + x + 5 + " ");**

**System.out.println(x + y + foo());**

**}**

**static String foo() { return "foo"; }**

**}**

**What is the result?**

A. 9 foo47 86foo

B. 9 foo47 4244foo

C. 9 foo425 86foo

D. 9 foo425 4244foo

E. 72 foo47 86foo

F. 72 foo47 4244foo

G. 72 foo425 86foo

H. 72 foo425 4244foo

I. Compilation fails.

**Correct Option : G**

**12. Place the fragments into the code to produce the output 33. Note, you must use each fragment exactly once.**

**CODE:**

**class Incr {**

**public static void main(String[] args) {**

**Integer x = 7;**

**int y = 2;**

**x \_\_\_ \_\_\_; x\*=x;**

**\_\_\_ \_\_\_ \_\_\_; y\*=y;**

**\_\_\_ \_\_\_ \_\_\_; y\*=y;**

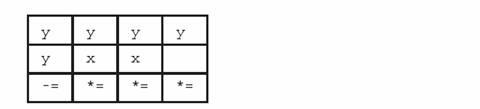
**\_\_\_ \_\_\_ \_\_\_; x-=y**

**System.out.println(x);**

**}**

**}**

FRAGMENTS:



ass Incr {

public static void main(String[] args) {

Integer x = 7;

int y = 2;

x \*= x;

y \*= y;

y \*= y;

x -= y;

System.out.println(x);

}

}

Yeah, we know it’s kind of puzzle-y, but you might encounter something like it on the real exam.

x \*= x;

y \*= y;

y \*= y;

x -= y;

**13. Given:**

**1. class Maybe {**

**2. public static void main(String[] args) {**

**3. boolean b1 = true;**

**4. boolean b2 = false;**

**5. System.out.print(!false ^ false);**

**6. System.out.print(" " + (!b1 & (b2 = true)));**

**7. System.out.println(" " + (b2 ^ b1));**

**8. }**

**9. }**

**Which are true?**

A. Line 5 produces true.

B. Line 5 produces false.

C. Line 6 produces true.

D. Line 6 produces false.

E. Line 7 produces true.

F. Line 7 produces false.

**Correct option: A,D,F**

**14. Given:**

**class Sixties {**

**public static void main(String[] args) {**

**int x = 5;**

**int y = 7;**

**System.out.print(((y \* 2) % x));**

**System.out.print(" " + (y % x));**

**}**

**}**

**What is the result?**

A. 1 1

B. 1 2

C. 2 1

D. 2 2

E. 4 1

F. 4 2

G. Compilation fails.

H. An exception is thrown at runtime.

**Correct Option: F**

**15. Given:**

**class Scoop {**

**static int thrower() throws Exception { return 42; }**

**public static void main(String [] args) {**

**try {**

**int x = thrower();**

**} catch (Exception e) {**

**x++;**

**} finally {**

**System.out.println("x = " + ++x);**

**} } }**

**What is the result?**

A. x = 42

B. x = 43

C. x = 44

D. Compilation fails.

E. The code runs with no output.

**Correct Option : D**

**16. Given:**

**class Alien {**

**String invade(short ships) { return "a few"; }**

**String invade(short... ships) { return "many"; }**

**}**

**class Defender {**

**public static void main(String [] args) {**

**System.out.println(new Alien().invade(7));**

**} }**

**What is the result?**

A. many

B. a few

C. Compilation fails.

D. The output is not predictable.

E. An exception is thrown at runtime.

**Correct Option: C**

**Invalid calling of method**

**17. Given:**

**1. class Dims {**

**2. public static void main(String[] args) {**

**3. int[][] a = {{1,2,}, {3,4}};**

**4. int[] b = (int[]) a[1];**

**5. Object o1 = a;**

**6. int[][] a2 = (int[][]) o1;**

**7. int[] b2 = (int[]) o1;**

**8. System.out.println(b[1]);**

**9. } }**

**What is the result?**

A. 2

B. 4

C. An exception is thrown at runtime

D. Compilation fails due to an error on line 4.

E. Compilation fails due to an error on line 5.

F. Compilation fails due to an error on line 6.

G. Compilation fails due to an error on line 7.

**18. Given:**

**class Eggs {**

**int doX(Long x, Long y) { return 1; }**

**int doX(long... x) { return 2; }**

**int doX(Integer x, Integer y) { return 3; }**

**int doX(Number n, Number m) { return 4; }**

**public static void main(String[] args) {**

**new Eggs().go();**

**}**

**void go() {**

**short s = 7;**

**System.out.print(doX(s,s) + " ");**

**System.out.println(doX(7,7));**

**} }**

**What is the result?**

A. 1 1

B. 2 1

C. 3 1

D. 4 1

E. 2 3

F. 3 3

G. 4 3

**Correct option : G**

**19. Given:**

**class Mixer {**

**Mixer() { }**

**Mixer(Mixer m) { m1 = m; }**

**Mixer m1;**

**public static void main(String[] args) {**

**Mixer m2 = new Mixer();**

**Mixer m3 = new Mixer(m2); m3.go();**

**Mixer m4 = m3.m1; m4.go();**

**Mixer m5 = m2.m1; m5.go();**

**}**

**void go() { System.out.print("hi "); }**

**}**

**What is the result?**

A. hi

B. hi hi

C. hi hi hi

D. Compilation fails

E. hi, followed by an exception

F. hi hi, followed by an exception

**Correct Option: F**

**20. Given:**

**1. class Zippy {**

**2. String[] x;**

**3. int[] a [] = {{1,2}, {1}};**

**4. Object c = new long[4];**

**5. Object[] d = x;**

**6. }**

**What is the result?**

A. Compilation succeeds.

B. Compilation fails due only to an error on line 3.

C. Compilation fails due only to an error on line 4.

D. Compilation fails due only to an error on line 5.

E. Compilation fails due to errors on lines 3 and 5.

F. Compilation fails due to errors on lines 3, 4, and 5.

**Correct Option: D**

**21. Given:**

**class Fizz {**

**int x = 5;**

**public static void main(String[] args) {**

**final Fizz f1 = new Fizz();**

**Fizz f2 = new Fizz();**

**Fizz f3 = FizzSwitch(f1,f2);**

**System.out.println((f1 == f3) + " " + (f1.x == f3.x));**

**}**

**static Fizz FizzSwitch(Fizz x, Fizz y) {**

**final Fizz z = x;**

**z.x = 6;**

**return z;**

**} }**

**What is the result?**

A. true true B. false true C. true false

D. false false E. Compilation fails. F. An exception is thrown at runtime.

**Explanation:**

**Correct Option: A**

**Because in FizzSwitch method it will return the object of “f1” and it will return to “f3” object.**

**22. Given:**

**class Knowing {**

**static final long tooth = 343L;**

**static long doIt(long tooth) {**

**System.out.print(++tooth + " ");**

**return ++tooth;**

**}**

**public static void main(String[] args) {**

**System.out.print(tooth + " ");**

**final long tooth = 340L;**

**new Knowing().doIt(tooth);**

**System.out.println(tooth);**

**} }**

**What is the result?**

A. 343 340 340

B. 343 340 342 C. 343 341 342

D. 343 341 340 E. 343 341 343 F. Compilation fails.

G. An exception is thrown at runtime.

**Correct Option : D**

**23. Given:**

**1. class Bigger {**

**2. public static void main(String[] args) {**

**3. // insert code here**

**4. }**

**5. }**

**6. class Better {**

**7. enum Faster {Higher, Longer};**

**8. }**

**Which, inserted independently at line 3, will compile? (Choose all that apply.)**

A. Faster f = Faster.Higher;

B. Faster f = Better.Faster.Higher;

C. Better.Faster f = Better.Faster.Higher;

D. Bigger.Faster f = Bigger.Faster.Higher;

E. Better.Faster f2; f2 = Better.Faster.Longer;

F. Better b; b.Faster = f3; f3 = Better.Faster.Longer;