Find Output

```
int x = 1;
if (x) {
        System.out.println("hello");
}
else{
        System.out.println("bai");
}
```

a) hello
b) bai
c) No output
d) Compilation error

Ans:

d

2. The …… statement tests the value of a given variable against a list of case values and when a match is found, a block of statements associated with that case is executed.

A) switch

B) break

C) continue

D) default

Ans: A

3. …….. statement provides an easy way to dispatch execution to different parts of your code based on the value of an expression.

A) if-else

B) switch

C) if

D) while

Ans: B

4. State whether the following statements about switch statement are correct.

i) Switch statement often provides a better alternative than a large series of if-else-if statements.

ii)The break statement is used inside the switch to terminate a statement sequence.

A) True, False

B) False, True

C) True, True

D) False, False

Ans: C

5. The conditional statement, ……. can only test for equality, whereas ……. can evaluate any type of Boolean expression.

A) if, switch

B) switch, if

C) while, if

D) if, while

Ans: B

6. What will be the output of the following code snippet?

```
int a=15;
int b=25;
if ((a<b) || (a=5)>15)
System.out.println(a);
else
System.out.println(b);
```

A) Error

B) 15

C) 25

D) No output

Ans: B

Find output

```
int a=10,b=20;
    if(a<b || ++a<25)
    System.out.println(a);
    if(a>b || ++a<25)
    System.out.println(a);
```

- Output:

10

11

Find output

```
int a=10,b=20;
    if(a<b && ++a<25)
    System.out.println(a);
    if(a>b && ++a<25)
    System.out.println(a);
```

Output:

11

7. What will be the output of the following code.

```
 int x, y;
x=15; y=20;
if (x>15)
if(y>15)
{
System.out.println(“y is  “+y);
}
else
System.out.println(“x is “+x);
```

A) Error

B) y is 20

C) x is 15

D) No output

Ans: D

8. In while and do-while loops, a …… statement causes control to be transferred directly to the conditional expression that controls the loop.

A) break

B) pause

C) start

D) continue

Ans: D

9. State whether the following statements about switch statement are True or False.

i) No two case constants in the same switch can have identical values.

ii) A switch statement is usually more efficient than a set of nested ifs.

A) True, False

B) False, True

C) True, True

D) False, False

Ans: C

11. Which of the following control expressions are valid for an if statement?

A) an integer expression

B) a Boolean expression

C) either A or B

D) Neither A or B

Ans: B

12. In the following code snippet, which lines of code contain error?

```
1. int j=0;
2. while (j<10){
3. j++;
4. if (j==5) continue loop;
5. system.out.println(“j is ” +j);}
```

A) Line 2

B) Line 3

C) Line 4

D) Line 5

Ans: C

13. State output of the following code

```
int x=20;
int y=10;
if(x>y)
{
if (y>10)
{
System.out.println(“y is “+y);
}
else
System.out.println(“x is  “+x);  }
```

A) Error

B) x is 20

C) y is 10

D) No output

Ans: B

14. By using ……., you can force immediate termination of loop, bypassing the conditional expression and any remaining code in the body of the loop.

A) switch

B) break

C) continue

D) default

Ans: B

15. The …… loop is especially useful when you process a menu selection.

A) while

B) do-while

C) for

D) switch

Ans: B

16. If you need to select among a large group of values, a switch statement will run much faster than the equivalent logic coded using ……. statement.

A) if

B) if-else

C) do-while

D) while

Ans: B

17. What will be the output of the following code.

```
int j=50;
while(true)
{
if(j<10)
break;
j=j-10;
}
system.out.println(“j is  “+j);
```

A) Error

B) j is 0

C) j is 50

D) No output

Ans: B

18. Here is a segment of a program

```
x=1;
y=1;
if(n>0)
x=x+1;
y=y-1;
```

what will be the values of x and y if n=1.

A) x=1, y=1

B) x=0, y=2

C) x=2, y=1

D) x=2, y=0

Ans: D

Find output

```
if(true);
System.out.println("hello ");
System.out.println("java ");
```

hello

java

Find output

```
int i = 0, j = 9;
do {
        i++;
        if (j-- < i++) {

                break;
        }
} while (i < 5);
System.out.println(i + "" + j);
```

Options:
A) 44
B) 55
C) 66
D) 77

Ans:

C

Find output

```
int j = 0;
do
        for (int i = 0; i++ < 1;)
                System.out.print(i);
while (j++ < 2);
```

Options:

1. 111
2. 222
3. 333
4. error

Ans: 1

```
class Test {
        static String s = "";
Public static void main(String[] args)
        {
        P:
                for (int i = 2; i < 7; i++) {
                        if (i == 3)
                                continue;
                        if (i == 5)
                                break P;
                        s = s + i;
                }
                System.out.println(s);
        }
}
```

Options:

1. 32
2. 23
3. 24
4. 42

Ans:

3

Find Output

```
class Test {
public static void main(String[] args)
   {
      int x = 10;
      if (++x < 10 && (x / 0 > 10)) {
         System.out.println(“raju");
      } else {
         System.out.println(“rani");
      }
   }
}
```

Options:

1. Compile time error
2. RuntimeException:ArithmeticException: / by zero
3. raju
4. rani

Ans:

4

Find Output

```
public class Main {
public static void main(String[] args) {
 int x = 10;
        if ((x / 0 > 10) &&  ++x < 10 ) {
            System.out.println("raju");
        } else {
            System.out.println("rani");
        }     }
}
```

Options:

1. Compile time error
2. RuntimeException:ArithmeticException: / by zero
3. raju
4. rani

Ans:

2

Find Output

```
Public class Test {
Public static void main(String[] args)
    {
        int b = 2147483648;
        System.out.println(b);
    }
}
```

Options:

1. No output
2. 2147483648
3. 2147483647
4. compile-time error

Ans:

4

Find Output

```
public class Main {
static char ch = 68;
    public static void main(String[] args)
    {
        System.out.println(ch);
    }
}
```

Options:

1. compile-time error
2. null
3. No output
4. D

Ans:

4

Find Output

```
public class Main {
public static void main(String[] args)
    {
    int x = 0xRaju;
        System.out.println(x);
    }
}
```

Options:

1. 1
2. Compile-time error
3. null
4. Run-time error

Option
2

Find Output

```
public class Main {
    public static void main(String[] args)
    {
    int x = 10F;
        System.out.println(x);
    }
}
```

Options:

1. 10
2. Compile-time error
3. 10.0
4. RuntimeException

Ans:

2

Find Output

```
class Test {
public static void main(String[] args)
   {
      do
         while (true)
            System.out.println("HELLO");
      while (false);
   }
}
```

Options:

1. HELLO
2. Compile time error
3. HELLO (infinitely)
4. No Output

option :

3

Find output

```
public class Main {
    public static void main(String[] args)
    {
    do
            System.out.println("FRIENDS");
        while (true);
        System.out.println("ENEMY");
    }
}
```

Options:

1. Compile time error
2. FRIENDS
3. No output
4. ENEMY

option

(1)

Find Output

```
public class Main {
    public static void main(String[] args)
    {
    while(true)
            System.out.println("FRIENDS");
        System.out.println("ENEMY");
    }
}
```

Options:

1. Compile time error
2. FRIENDS
3. No output
4. ENEMY

option

(1)

Find Output

```
public class Main {
    public static void main(String[] args)
    {
    final int a = 10, b = 20;
    while(a<b)
            System.out.println("FRIENDS");
        System.out.println("ENEMY");
    }
}
```

Options:

1. Compile time error
2. FRIENDS
3. No output
4. ENEMY

option

(1)

Find Output

```
class Test {
Public static void main(String[] args)
   {
      int x = 1, y = 2;
      do
         System.out.println("FRIENDS");
      while (x < y);
      System.out.println("ENEMY");
   }
}
```

Options:

1. FRIENDS
2. ENEMY
3. No Output
4. FRIENDS (Infinitely)

option (4)

Find Output

```
class Test {
public static void main(String[] args)
        {
                do
                        while (true)
                                ;
                System.out.println("HELLO");
        }
}
```

Options:

1 HELLO

2 HELLO (Infinitely)

3 Error: Unreachable statement

4 Error: ; expected

option

(4)

Find Output

```
class Test {
public static void main(String[] args)
   {
      do {
         System.out.print(1);
         do {

            System.out.print(2);

         } while (false);

      } while (false);
   }
}
```

Options:

1. 12
2. 21
3. 1
4. 2

option (1)

Find Output

```
Public class Test {
Public static void main(String[] args)
    {
        for (;;)
            System.out.println(“java");
    }
}
```

Options:

1. java
2. Compile time error
3. Run time Exception
4. java (Infinitely)

option (4)

Find Output

```
class Test {
public static void main(String[] args)
    {
        boolean b = true;
        if (b = false) {
            System.out.println("HELLO");
        } else {
            System.out.println("BYE");
        }
    }
}
```

Option:

1. HELLO
2. BYE
3. Compile time error: re- initialization
4. No output

option

(2)

Find Output

```
Public class Test {
Public static void main(String[] args)
   {
      int a = 10, b = 20;
      if (a < b) {
         if (a > b) {
            System.out.println("HELLO RAJU");
         } else {
            System.out.println("WELCOME");
         }
      }
   }
}
```

Option:

1.HELLO RAJU

2.WELCOME

3.Compile time error

4.HELLO RAJU WELCOME

option (2)

Find Output

```
class Test {
public static void main(String[] args)
    {
        for (int i = 0;; i++) {
            System.out.println("HIII");
        }
        System.out.println("BYE");
    }
}
```

Options:

1. HIII
2. HIII(infinitely)
3. BYE
4. Compile time error

option (4)

Find output

```
 boolean value = true;
if (!value)
System.out.println("True");
else
System.out.println("False");
```

Output

False

Find output

```
int marks = 40;
switch(marks > 45) {
case true: System.out.println("PASS");
                    break;
case false: System.out.println("FAIL");
                    break;
```

Output

- Compilation error

Find output

```
boolean b = false;
if (b = true) {
System.out.println("b is true");
 }
else {
System.out.println("b is false");
}
```

Output

- b is true

Find output

```
int x=10;
switch(x)
{
        case 5: System.out.println("5");
        case 10: System.out.println("10");
        case 15: System.out.println("15");
                 break;
        default: System.out.println("default case");
}
```

Output

10

15

Find output

```
int array_variable [] = new int[5];
for (int i = 0; i < 5; i++) {
        System.out.print(array_variable[i] + " ");
    }
```

Output

0 0 0 0 0

Find output

```
int array_variable [] = new int[10];
for (int i = 0; i < 10; ++i) {
        array_variable[i] = i;
        System.out.print(array_variable[i] + " ");
        i++;
    }
```

Output

0 2 4 6 8

Find output

```
int num = 0;
while(++num < 4) {
System.out.print(num + " ");
}
```

Output

1 2 3

Find output

```
int begin = 10;
int end = 15;
do {
begin++;
System.out.print(begin + " ");
}
while(begin < end);
```

Output

11 12 13 14 15

Find output

```
int num = (10 <  20)? 10 : 20;
System.out.println(num);
```

Output

10

Find output

```
int num = 1;
for( ; ; ) {
        System.out.println(num);
        if(num < 5) {
                break;
        }
}
```

Output

1

Find output

```
int start = 10;
while(start == 10) {
System.out.println(start);
start++;
}
```

Output

10

Find output

```
int num = 1;
switch(num){
case 1: num++;
case 2: num++;
case 0: num++;
default : num++;
   }
System.out.println(num);
```

Output

5

Find output

```
int a,b=3;
a=b;
if(a>b)
System.out.println("a is big");
else
System.out.println("b is big");
```

Output

b is big

Find output

```
int i;
for(i=3;i<15;i+=3)
        System.out.println(i);

System.out.println("------------");
for(i=3;i<15;i+=3);
        System.out.println(i);
```

Output

3

6

9

12

15

Find output

```
for (;;)
        System.out.println(“raju");
```

Output

Options:
1.raju
2.Compile time error
3.Run time Exception
4.raju (Infinitely)

Ans: 4

Explanation: In the above example, we are using for loop. In for loop if we did not provide any initialization, condition-check and increment/decrement part then it will go to infinite loop if we did not provide any condition in statement.

Find output

```
boolean b = true;
if (b = false) {
        System.out.println("HELLO");
} else {
        System.out.println("BYE");
      }
```

Output

1. HELLO
2. BYE
3. Compile time error: re- initialization
4. No output

- The answer is option (2)
- **Explanation :** In the condition of if statement, we assigning are false to b which return a boolean value which is false. Therefore the control goes to the else part and the output is BYE.