1) class increment {
Public static void main (String args [])

{

int g = 3;

System.out.print(++g *8);

}

}

→ g = 3

++g = 4

(++g * 8) = 4 x 8

= 32

O/P = 32

2) class output {
Pubic static void main (String args[])

{

double a,b,c;

a = 3.0/0;

b = 0/4.0;

c = 0/0.0;

System.out.println(a);

System.out.println(b);

System.out.println(c);

}

}

O/P = NaN

3) class variable – scope

```
{
Public static void main (String args[])
{
 int x;
 x = 5;
 {
 int y = 6;
System.out.print (x +" " + y);
 }
System.out.print (x +" " +y);
}
}
```

O/P ⇒ Compilation error

cannot find y in variable

4) what will be the error in the following Java code?

```
byte b = 50;
b = b * 50
```

O/P ⇒ * operator has converted b* 50 into int, which can be converted to byte without Casting.

5) class evaluate

```
{
Public static void main (String args [])
{
int arr [] = new int [] {0,1,2,3,4,5,6,7,8,9};
int n = 6
n=arr [arr [n]/2 ];
System.out. println (arr [n]/2);
}
}
```

O/P → 1

n = arr [arr [n]/2]

n = arr[6/2]

n = arr [3]

n = 2

= arr[n]/2

= 2/2

= 1

6) class leftshift_operator

```
{
Public static void main (String args[])
 {
byte x = 64;
 int i;
byte y;
i = x << 2;
y = (byte)(x << 2)
System.out.print(i + " " + y);
}
}
```

O/P → 256 0

x = 64
i = x << 2 : 4
64 x 4 = 256 + x

byte rang
∴ y = 0

7) What will be the output

```
class output
 {
Public static void main(String args[])
 {
 int x = y = z = 20;
 }
}
```

O/P → compile time error

y and x are not initialize.

8) class box
{
int width;
int height;
int length;
}
class mainclass
{
Public static void main (string args[])
{
box obj = new box();
obj.width = 10;
obj.height = 2;
obj.length = 10;
int y = obj.width * obj.height * obj.length;
System.out.print(y);
}
}

O/P → 200

y = 10 x 2 x 10
y = 200

9) class output
{
Pubic static void main (String args[])
{
int arr[] = {1,2,3,4,5};
for { int i = 0; i < arr.length - 2; ++i)
System.out.println (arr[i] + " ");
}
}

O/P → 1 2 3
arr.length = 5 (5-2=3)

∴ 1, 2, 3

10) class abc
{
Public static void main (String args[])
{
if (args.length > 0)
System.out.println (args.length);
}
}

o/p → The snippet compiles and runs but does not print anything

args is 0 so the code run but not output is printed.

11) Class Alligator
{
Public static void main (String [] args)
{
int [] x [] = {{1,2}, {3,4,5}, {6,7,8,9}};
int [][] y = x;
System.out.println {y[2] [1]};
}
}

o/p → 7

12) class A

```
{
int i;
void display()
{
System.out.println(i);
}
}
class B extends A
{
int j;
void display()
{
System.out.println(j);
}
}
class method_overriding.
{
public static void main (string args[])
{
B obj = new B();
obj.i = 1;
obj.j = 2;
obj.display();
}
}
```

O/P → 2

B and A both has display() but B inherit A ∴ B display() exectes.

13)

```
class String_demo
 {
Public static void main (String args[])
 {
 Char chars[] = {'a', 'b', 'c'};
 String s = new string(Chars);
  System.out.println(s);
   }
  }
```

O/P → abc

initialized string s

∴ abc is execute.

14) class recursion
```
{
int func(int n)
{
int reasult;
if (n==1)
 return 1;
result = func(n-1);
 return result;
 }
}
Class output
{
Public.static void main (String args[])
{
recursion obj = new recursion();
System.out.print(obj.func(5));
  }
}
```

O/P → 1

15) class output
```
{
 Public static void main (String args[])
  {
 String c = "Hello i love java";
  boolean var;
 Var = c.starts with ("hello");
  System.out.println(var);
  }
 }
```

O/P → false

starts with() is a case sensitive.

16) class output{
Public static void main (String args[])
{
StringBuffer s1 = new StringBuffer ("Hello");
String Buffer s2 = s1. reverse();
System. out. println (s2);
}
}

O/P → olleH

reverse() method reverse the string

17) class output {
Public static void main(String args[])
{
Integer i = new interger (257);
byte x = i. byteValue();
System. out. print (x);
}
}

O/P → 1

byte length is -256 to 256

∴ 1 is printed

18)

```
Class output
{
Public static void main (String args [])
{
 double x = 2.0;
 double Y = 3.0;
 double z = Math.pow(x,y);
 System.out.print(z);

 }
}
```

O/P → 8.0

$$2^3 = 8$$

19)

```
class output
{
Pubic static void main (String args [])

 {
 double x = 3.14;
 int y = (int) Math.ceil (x);
 System.out.print(Y);
  }
}
```

O/P → 4

Print whole no greater or equal to x and its also initiazied int ∴ 4 is executed.

20)

```
class output
{
Pubic static void main (String args[])
{
 int a = Character.MIN_VALUE;
 System.out.print((char)a);
 }
}
```

O/P → Space.

smallest character value is ' '
Space.

21)

```
Class networking
{
Public static void main (String [] args) throws Exception
 {
URL obj = new URL ("https://www.google.com");
URLConnection obj1 = obj.openConnection();
int len = obj1.getContentLength();
System.out.Print(len);
 }
}
```

O/P → 127

22)
```
class networking
{
Public static void main (String [] args) throws Malformed
URLException
{
URL obj = new URL ("http://www.google.com");
System.out.print(obj.toExternalForm());
}
}
```

o/p → https://google.com.

23)
```
class Arraylist
{
Public static Void main (String args[])
{
ArrayListobj = new ArrayList();
obj.add("A");
obj.add("B");
obj.add("c");
obj.add("D");
System.out.println(obj);
}
}
```

o/p → [A, B, C, D]

array is added and in last at position of index 1, D is added. A, B, C, D is printed.

24)

```
Class date
{
Public static void main (String args [])
 {
 Data obj = new Date ();
 System.out.print(obj);
 }
}
```

O/P → Prints present Time and Date.

25)

```
Class Collection_iterators
 {
Public static void main (String args [])
 {
 LinkedList list = new LinkdList();
 list.add (new Integer (2));
 list.add (new Integer(8));
 list.add (new Integer(5));
 list.add (new Integer(1));
 iterator i = list.iterator();
 Collection.reverse (list);
 Collection.sort(list);
 while (i.hasNext())
System.out.print (i.next()+ " ");
 }
}
```

O/P → 1, 2, 5, 8

collection sort () sorts the list 1258

26)
```
class multithreaded_programming
{
Public static void main (String args [])
{
 Thread t = Thread . currentThread ();
  t . setName ("Name Thread");
  System . out . println (t);
 }
}
```

O/P → Thread [New Thread, 5, main]

27)
```
class output
{
Pubic static void main (String args [])
{
 StringBuffer S1 = new StringBuffer ("Hello");
 StringBuffer S2 = S1 . reverse ();
 System . out . println (S2);
 }
}
```

O/P → olleH

reverse () method reverse the string.

28)

```
class newthread extends Thread
{
Thread t;
newthread( )
 {
t1 = new Thread(this.*Thread_1*);
t2 = new Thread(this.*Thread_2*);
 t1.start( );
 t2.start( );
 }
  Public void run( )
  {
t2.setPriority(Thread.MAX_PRIORITY);
  System.out.print(t1.equals(t2));
 }
}
class multithreaded_programming.
{
 Public static void main(String args[])
  {
 new heewthread( );
 }
}
```

O/P ⟶ falsefalse

29)

```
class overload
{
int x;
double y;
void add (int a, int b)
{
x = a+b;
}
void add (double c, double d)
{
y = c+d;
}
overload ()
{
  this.x = 0
  this.y = 0;
}}
class Overload_methods
{
Public static void main (String args[])
{
overload obj = new overload ();
  int a = 2;
double b = 3.2;
  obj.add (a,a);
  obj.add (b,b);
System.out.println (obj.x+" "+obj.y);
}
}
```

O/P → 4, 6.4

30)

```
Class string_class
{
Public static void main (String args[])
 {
  String obj = "I LIKE JAVA";
  System.out.println(obj.length());
 }
}
```

O/P → 12

31)

```
class array_output
 {
Public static void main (String args[])
 {
  int array_varible [] = new int [10];
   for (int i = 0; i<10; ++1)
 {
array-Variable [i] = i;
System.out.print (array_varible [i] + " ");
  i++;
 }
}
```

O/P ⇒

32)

```
class multidimention_array
{
Public static void main(String args[])
{
int arr[][] = new int[3][];
 arr[0] = new int[1];
 arr[1] = new int[2];
 arr[2] = new int[3];
 int sum = 0;
 for (int i = 0; i<3; ++i)
 for (int j = 0; j<i+1; ++j)
 arr[i][j] = j + 1;
 for (int i = 0; i<3; ++j)
 arr[i][j] = j+1;
 for(int i =0; i<3; ++i)
 for(int j = 0; j<i+1; ++j)
  sum+= arr[i][j];
 System.out.print(sum);
}
}
```

o/p → 10

in 2D array 1 row is 1 ele 2 row is 2 ele and 3 row is 3 ele.

33) class array-output

```
{
Pubic static void main (String args[])
  {
  char array-variable[] = new char[10];
  for (int i=0; i<10; ++i)
   {
   array-variable[i] = '1';
  System.out.print(array-variable[i]+"");
    }
  }
}
```

O/P → 1 1 1 1 1 1 1 1 1 1

34) class array_output

```
{
Public static void main (String args [])
{
int array_Variable [][] = {{1,2,3},{4,5,6},
    {7,8,9}};
 int sum = 0;
 for (int i = 0; i < 3; ++i)
 for (int j = 0; j < 3; ++j)
 sum = sum + array_variable [i][j];
 System.out.print (sum/5);
}
}
```

O/P → 9

∴ i = 2 , j = 2

∴ array variable [i][j] = 9.

35) class box

```
{
int width;
int height;
int length;
int Volume;
void Volume(int height, int length, int width)
{
Volume = width * height * length;
}
}
class Parameterized_method {
Pubic static void main(String args[])
{
box obj = new box();
obj.height = 1;
obj.length = 5;
obj.width = 5;
obj Volume = (3,2,1);
System.out.println(obj.volume);
}
}
```

O/P → 6.

36) class equality

```
{
int x;
int y;
boolean isequal()
{
return(x==y);
}
}
class output
{
pubic static void main(String args[])
{
equality obj = new equality();
obj.x=5;
obj.y=5;
System.out.println(obj.isequal);
}
}
```

o/p = → True.

new obj is equal.

37) class box

```
{
int width;
int height;
int length;
int volume;
Void volume ()
{
Volume = width * height * length;
}
Void volume (int x)
{
Volume = x;
}
}
Class Output
{
Public static void main (String args[])
{
box obj = new box ();
obj.height = 1;
obj.length = 5;
obj.Width = 5;
obj.volume (5);
System.out.println(obj.voume);
}
}
```

O/P → 5

38) class output

```
{
Static void main (String args[])
 {
  int x, y=1;
   x=10;
   if (x!=10 && x/0==0)
    System.out.println(y);
    else
    System.out.println(++y);
    }
}
```

o/p → Compilation Error

main method not found

Pubic is missing.

39)
```
class area
{
int width;
int length;
int height;
 area()
 {
  width = 5;
  length = 6;
  height = 1;
 }
 void volume()
 {
Volume = width * height * length;
 }
}
class cons_method
{
Public static void main(String args[])
{
area obj = new area();
 obj.volume();
System.out.println(obj.volume);
 }
}
```

O/P ⟶ 30

40) class recursion

```
{
int func (int n)
{
 int result;
 result = func (n-1);
  return result;
  }
}
Class output
{
Public static void main (String args
 {
recursion obj = neco recursion();
 System.out.print (obj.func(12));
  }
}
```

o/p → Runtime Error

Infinite loop occurs.