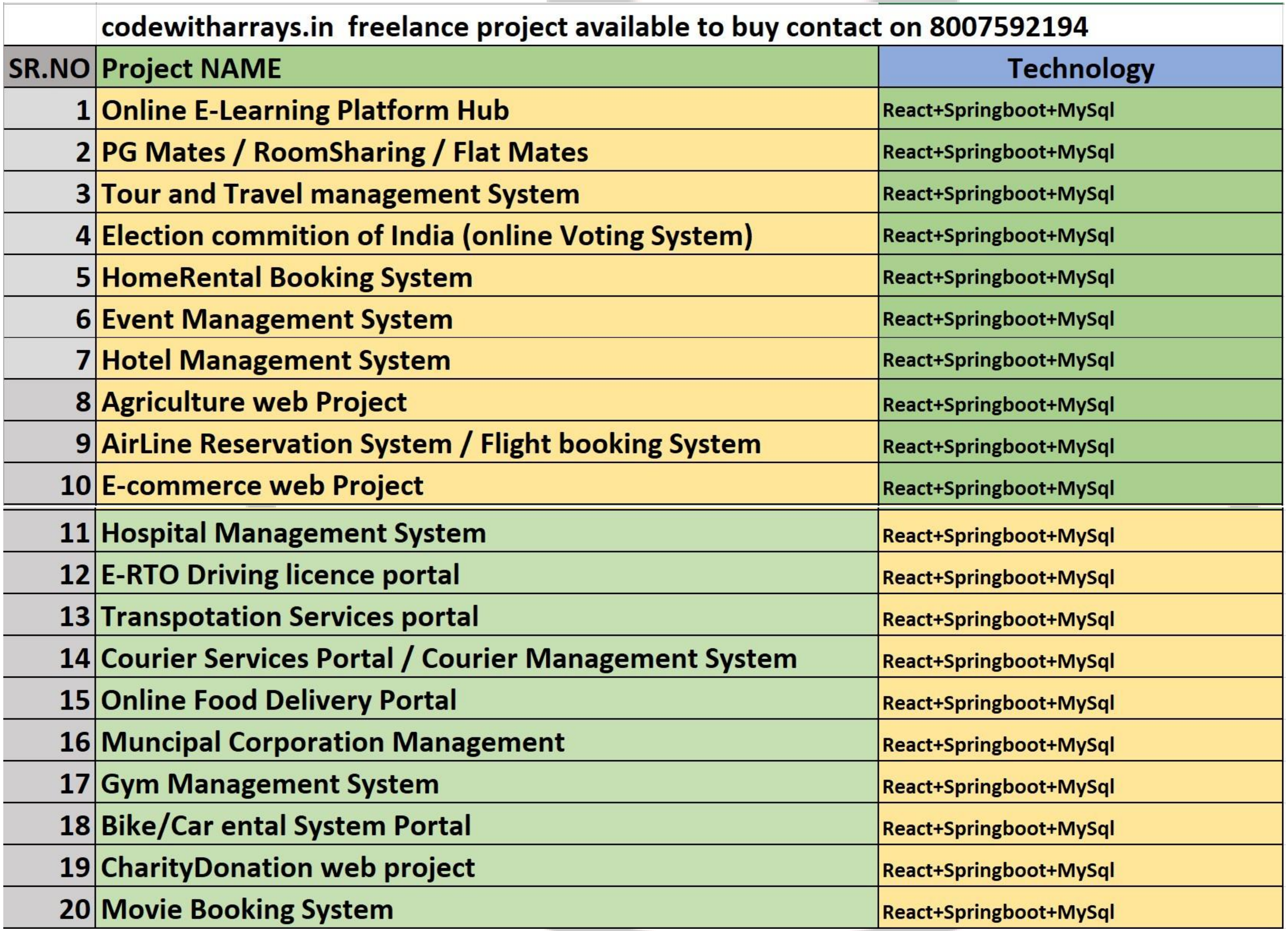
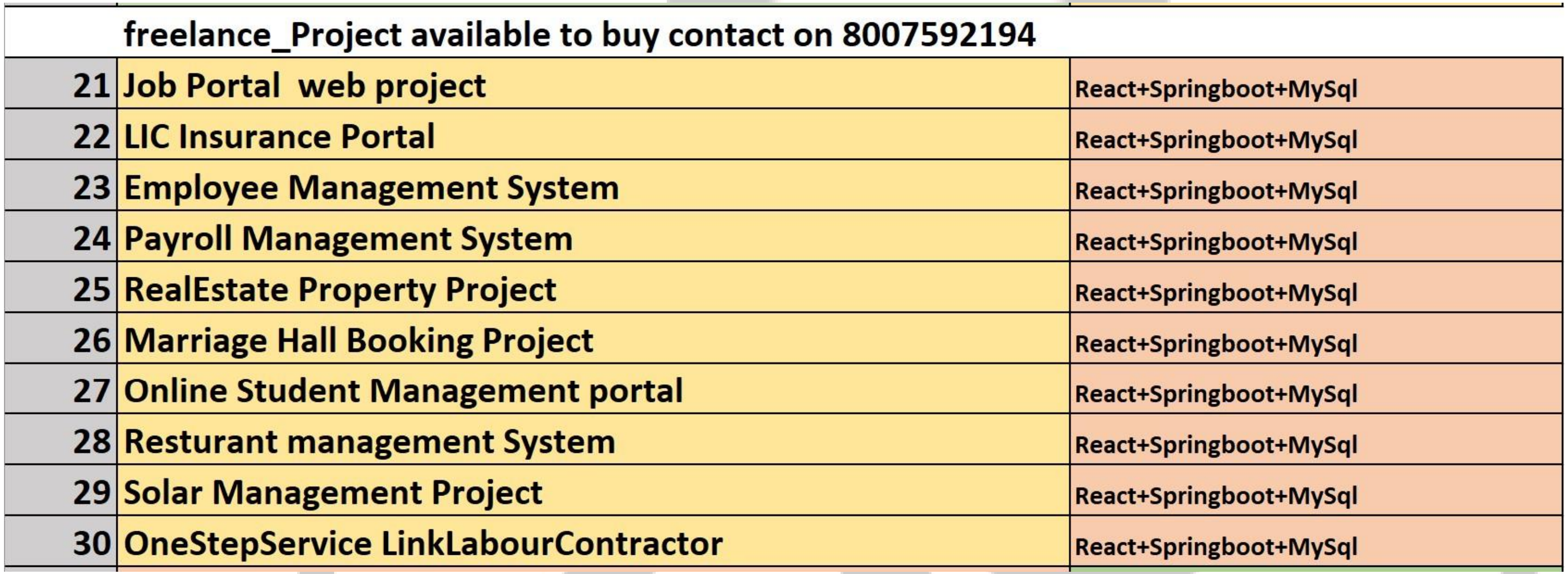
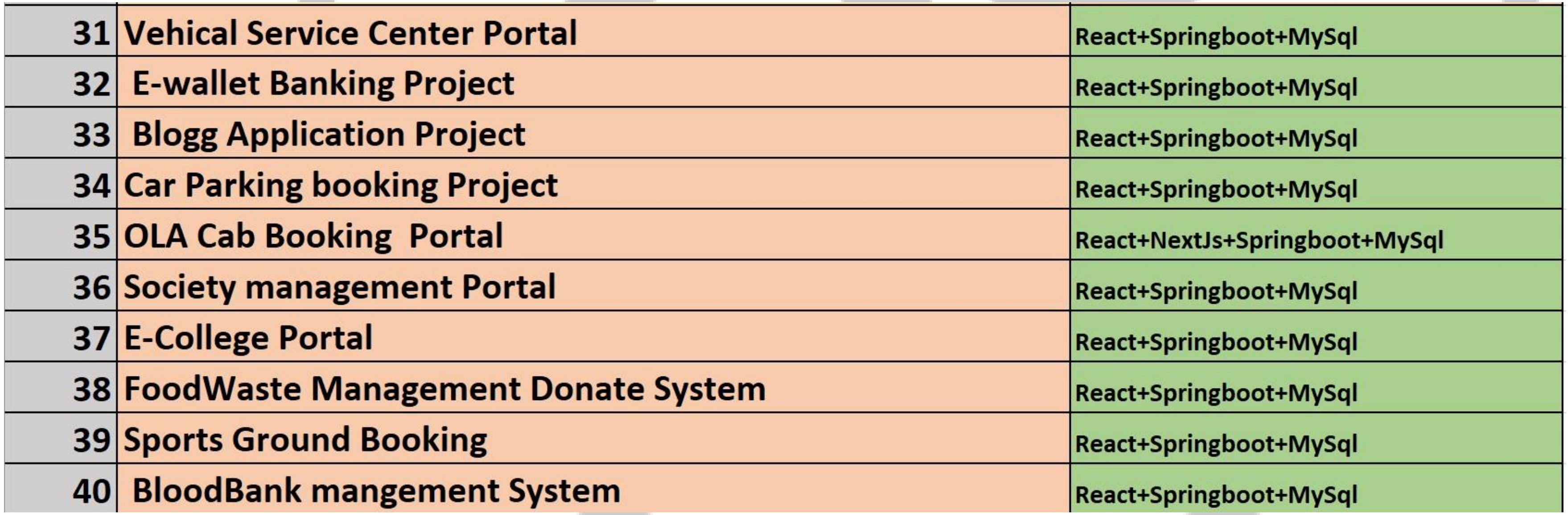


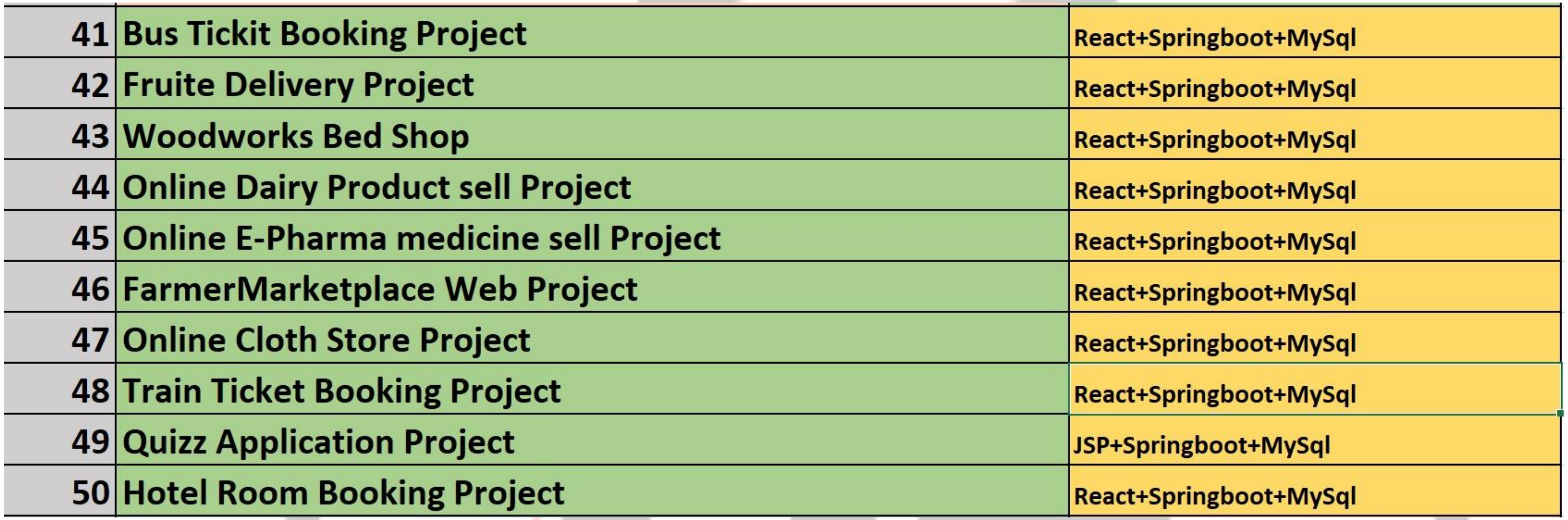
	codewitharrays.in freelance project available to buy contact on 8007592194	
SR.NO	Project NAME	Technology
1	Online E-Learning Platform Hub	React+Springboot+MySql
2	PG Mates / RoomSharing / Flat Mates	React+Springboot+MySql
3	Tour and Travel management System	React+Springboot+MySql
4	Election commition of India (online Voting System)	React+Springboot+MySql
5	HomeRental Booking System	React+Springboot+MySql
6	Event Management System	React+Springboot+MySql
7	Hotel Management System	React+Springboot+MySql
8	Agriculture web Project	React+Springboot+MySql
9	AirLine Reservation System / Flight booking System	React+Springboot+MySql
10	E-commerce web Project	React+Springboot+MySql
11	Hospital Management System	React+Springboot+MySql
12	E-RTO Driving licence portal	React+Springboot+MySql
13	Transpotation Services portal	React+Springboot+MySql
14	Courier Services Portal / Courier Management System	React+Springboot+MySql
15	Online Food Delivery Portal	React+Springboot+MySql
16	Muncipal Corporation Management	React+Springboot+MySql
17	Gym Management System	React+Springboot+MySql
18	Bike/Car ental System Portal	React+Springboot+MySql
19	CharityDonation web project	React+Springboot+MySql
20	Movie Booking System	React+Springboot+MySql



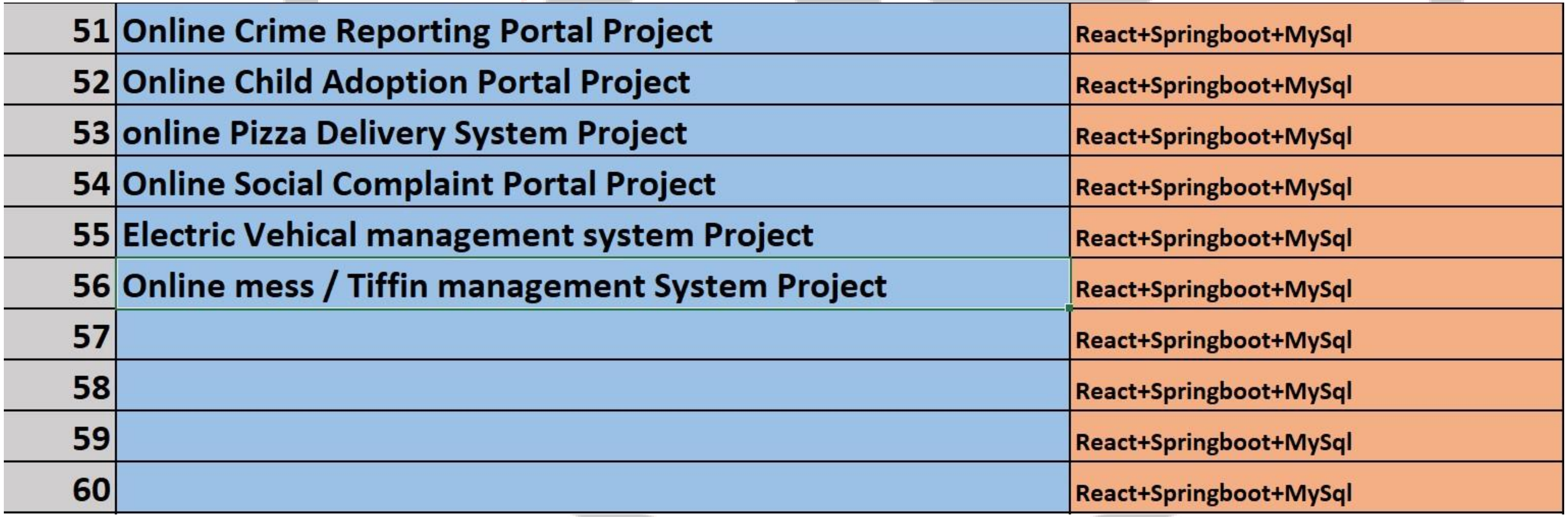
	freelance_Project available to buy contact on 8007592194	
21	Job Portal web project	React+Springboot+MySql
22	LIC Insurance Portal	React+Springboot+MySql
23	Employee Management System	React+Springboot+MySql
24	Payroll Management System	React+Springboot+MySql
25	RealEstate Property Project	React+Springboot+MySql
26	Marriage Hall Booking Project	React+Springboot+MySql
27	Online Student Management portal	React+Springboot+MySql
28	Resturant management System	React+Springboot+MySql
29	Solar Management Project	React+Springboot+MySql
30	OneStepService LinkLabourContractor	React+Springboot+MySql



31	Vehical Service Center Portal	React+Springboot+MySql
32	E-wallet Banking Project	React+Springboot+MySql
33	Blogg Application Project	React+Springboot+MySql
34	Car Parking booking Project	React+Springboot+MySql
35	OLA Cab Booking Portal	React+NextJs+Springboot+MySql
36	Society management Portal	React+Springboot+MySql
37	E-College Portal	React+Springboot+MySql
38	FoodWaste Management Donate System	React+Springboot+MySql
39	Sports Ground Booking	React+Springboot+MySql
40	BloodBank mangement System	React+Springboot+MySql



41	Bus Tickit Booking Project	React+Springboot+MySql
42	Fruite Delivery Project	React+Springboot+MySql
43	Woodworks Bed Shop	React+Springboot+MySql
44	Online Dairy Product sell Project	React+Springboot+MySql
45	Online E-Pharma medicine sell Project	React+Springboot+MySql
46	FarmerMarketplace Web Project	React+Springboot+MySql
47	Online Cloth Store Project	React+Springboot+MySql
48	Train Ticket Booking Project	React+Springboot+MySql
49	Quizz Application Project	JSP+Springboot+MySql
50	Hotel Room Booking Project	React+Springboot+MySql



51	Online Crime Reporting Portal Project	React+Springboot+MySql
52	Online Child Adoption Portal Project	React+Springboot+MySql
53	online Pizza Delivery System Project	React+Springboot+MySql
54	Online Social Complaint Portal Project	React+Springboot+MySql
55	Electric Vehical management system Project	React+Springboot+MySql
56	Online mess / Tiffin management System Project	React+Springboot+MySql
57		React+Springboot+MySql
58		React+Springboot+MySql
59		React+Springboot+MySql
60		React+Springboot+MySql

Spring Boot + React JS + MySQL Project List

Sr.No	Project Name	YouTube Link
1	Online E-Learning Hub Platform Project	https://youtu.be/KMjyBaWmgzg?si=YckHuNzs7eC84-IW
2	PG Mate / Room sharing/Flat sharing	https://youtu.be/4P9cIHg3wvk?si=4uEsi0962CG6Xodp
3	Tour and Travel System Project Version 1.0	https://youtu.be/-UHOBywHaP8?si=KHHfE_A0uv725f12
4	Marriage Hall Booking	https://youtu.be/VXz0kZQi5to?si=llOS-QG3TpAFP5k7
5	Ecommerce Shopping project	https://youtu.be/vJ_C6LkhrZ0?si=YhcBylSErvdn7paq
6	Bike Rental System Project	https://youtu.be/FIzsAmIBCbk?si=7ujQTJqEgkQ8ju2H
7	Multi-Restaurant management system	https://youtu.be/pvV-pM2Jf3s?si=PgvnT-yFc8ktrDxB
8	Hospital management system Project	https://youtu.be/IynIouBZvY4?si=CXzQs3BsRkjKhZCw
9	Municipal Corporation system Project	https://youtu.be/cVMx9NVyI4I?si=qX0oQt-GT-LR_5jF
10	Tour and Travel System Project version 2.0	https://youtu.be/_4u0mB9mHXE?si=gDiAhKBowi2gNUKZ

Sr.No	Project Name	YouTube Link
11	Tour and Travel System Project version 3.0	https://youtu.be/Dm7nOdpasWg?si=P_Lh2gcOFhlyudug
12	Gym Management system Project	https://youtu.be/J8_7Zrkg7ag?si=LcxV51ynfUB7OptX
13	Online Driving License system Project	https://youtu.be/3yRzsMs8TLE?si=JRI_z4FDx4Gmt7fn
14	Online Flight Booking system Project	https://youtu.be/m755rOwdk8U?si=HURvAY2VnizIyJlh
15	Employee management system project	https://youtu.be/lD1iE3W_GRw?si=Y_jv1xV_BljhrD0H
16	Online student school or college portal	https://youtu.be/4A25aEKfei0?si=RoVgZtxMk9TPdQvD
17	Online movie booking system project	https://youtu.be/Lfjv_U74SC4?si=fiDvrhhrjb4KSlSm
18	Online Pizza Delivery system project	https://youtu.be/Tp3izreZ458?si=8eWAOzA8SVdNwlyM
19	Online Crime Reporting system Project	https://youtu.be/0UlzReSk9tQ?si=6vN0e70TVY1GOwPO
20	Online Children Adoption Project	https://youtu.be/3T5HC2HKyT4?si=bntP78niYH802I7N

Java Inheritance Quiz Practice Coding Questions :

1) Tinku has written the code like below. But, it is showing compile time error. Can you identify what mistake he has done?

```
class X
{
    //Class X Members
}

class Y
{
    //Class Y Members
}

class Z extends X, Y
{
    //Class Z Members
}
```

View Answer

Answer :
In Java, a class can not extend more than one class. Class Z is extending two classes – Class X and Class Y. It is a compile time error in Java.

2) What will be the output of this program?

```
class A
{
    int i = 10;
}

class B extends A
{
    int i = 20;
}

public class MainClass
{
    public static void main(String[] args)
    {
        A a = new B();

        System.out.println(a.i);
    }
}
```

View Answer

Answer :
10

Java Inheritance Quiz

3) What will be the output of the below program?

```
class A
{
    {
        System.out.println(1);
    }
}

class B extends A
{
    {
        System.out.println(2);
    }
}

class C extends B
{
    {
        System.out.println(3);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();
    }
}
```

View Answer

Answer :
1
2
3

4) Can a class extend itself?

View Answer

Answer :
No. A class can not extend itself.

5) What will be the output of the following program?

```
class A
{
    String s = "Class A";
}

class B extends A
{
    String s = "Class B";

    {
        System.out.println(super.s);
    }
}

class C extends B
{
    String s = "Class C";

    {
        System.out.println(super.s);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();

        System.out.println(c.s);
    }
}
```

View Answer

Answer :
Class A

Class B
Class C

6) What will be the output of this program?

```
class A
{
    static
    {
        System.out.println("THIRD");
    }
}

class B extends A
{
    static
    {
        System.out.println("SECOND");
    }
}

class C extends B
{
    static
    {
        System.out.println("FIRST");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();
    }
}
```

View Answer

Answer :
THIRD
SECOND
FIRST

7) What will be the output of the below program?

```
class A
{
    public A()
    {
        System.out.println("Class A Constructor");
    }
}

class B extends A
{
    public B()
    {
        System.out.println("Class B Constructor");
    }
}
```

```
class C extends B
{
    public C()
    {
        System.out.println("Class C Constructor");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();
    }
}
```

View Answer

Answer :
Class A Constructor
Class B Constructor
Class C Constructor

8) Private members of a class are inherited to sub class. True or false?

View Answer

Answer :
false. Private members are not inherited to sub class.

9) What will be the output of the following program?

```
class X
{
    static void staticMethod()
    {
        System.out.println("Class X");
    }
}

class Y extends X
{
    static void staticMethod()
    {
        System.out.println("Class Y");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Y.staticMethod();
    }
}
```

View Answer

Answer :
Class Y

10) Below code is showing compile time error. Can you suggest the corrections?

```
class X
{
    public X(int i)
    {
        System.out.println(1);
    }
}

class Y extends X
{
    public Y()
    {
        System.out.println(2);
    }
}
```

View Answer

Answer :
Write explicit calling statement to super class constructor in Class Y constructor.

```
class X
{
    public X(int i)
    {
        System.out.println(1);
    }
}

class Y extends X
{
    public Y()
    {
        super(10);          //Correction

        System.out.println(2);
    }
}
```

11) What is wrong with the below code? Why it is showing compile time error?

```
public class A
{
    public A()
    {
        System.out.println(1);

        super();

        System.out.println(2);
    }
}
```

View Answer

Answer :
Constructor calling statements (super() or this()), if written, must be the first statements in the constructor.Correct Code…

```
public class A
{
    public A()
    {
        super();

        System.out.println(1);

        System.out.println(2);
    }
}
```

12) You know that compiler will keep super() calling statement implicitly as a first statement in every constructor. What happens if we write this() as a first statement in our constructor?

View Answer

Answer :
Compiler will not keep super() if you are writing this() as a first statement in your constructor.

13) Can you find out the error in the below code?

```
public class A
{
    public A(int i)
    {

    }
}

class B extends A
{

}
```

View Answer

Answer :
Class B doesn’t have any constructors written explicitly. So, compiler will keep default constructor. In that default constructor, there will be a calling statement to super class constructor (super()). But, it is undefined in Class A.To remove the errors, either define A() constructor in class A or write B() constructor in class B and call super class constructor explicitly.

```
public class A
{
    public A()
    {
        //Either keep this constructor
    }

    public A(int i)
    {

    }
}

class B extends A
{
    public B()
    {
        super(10);   //or else write this statement
    }
}
```

14) Which class is a default super class for every class in java?

View Answer

Answer :
java.lang.Object class

15) Can you find out the error in the below code?

```
public class A
{
    public A()
    {
        super();

        this(10);
    }

    public A(int i)
    {
        System.out.println(i);
    }
}
```

View Answer

Answer :
A constructor can have either super() or this() but not both.

16) What will be the output of this program?

```
class M
{
    static
    {
        System.out.println('A');
    }

    {
        System.out.println('B');
    }
    public M()
    {
        System.out.println('C');
    }
}

class N extends M
{
    static
    {
        System.out.println('D');
    }

    {
        System.out.println('E');
    }

    public N()
    {
        System.out.println('F');
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        N n = new N();
    }
}
```

View Answer

Answer :
A
D
B
C
E
F

17) What will be the output of the below program?

```
class M
{
    int i;

    public M(int i)
    {
        this.i = i--;
    }
}

class N extends M
{
    public N(int i)
    {
        super(++i);

        System.out.println(i);
    }
}
```

```
public class MainClass
{
    public static void main(String[] args)
    {
        N n = new N(26);
    }
}
```

View Answer

Answer :
27

18) What will be the output of the following program?

```
class M
{
    int i = 51;

    public M(int j)
    {
        System.out.println(i);

        this.i = j * 10;
    }
}

class N extends M
{
    public N(int j)
    {
        super(j);

        System.out.println(i);

        this.i = j * 20;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        N n = new N(26);

        System.out.println(n.i);
    }
}
```

View Answer

Answer :
51
260
520

19) Why Line 10 in the below code is showing compilation error?

```
class X
{
    private int m = 48;
}
```

```
class Y extends X
{
    void methodOfY()
    {
        System.out.println(m);
    }
}
```

View Answer

Answer :
Because, private field ‘m’ is not visible to class Y.

20) What will be the output of the below program?

```
class X
{
    int m = 1111;

    {
        m = m++;

        System.out.println(m);
    }
}

class Y extends X
{
    {
        System.out.println(methodOfY());
    }

    int methodOfY()
    {
        return m-- + --m;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Y y = new Y();
    }
}
```

View Answer

Answer :
1111
2220

21) What will be the output of this program?

```
class A
{
    static String s = "AAA";

    static
    {
        s = s + "BBB";
    }

    {
        s = "AAABBB";
    }
}

class B extends A
{
    static
    {
        s = s + "BBBAAA";
    }

    {
        System.out.println(s);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        B b = new B();
    }
}
```

View Answer

Answer :
AAABBB

22) What will be the output of the following program?

```
class X
{
    int i = 101010;

    public X()
    {
        i = i++ + i-- - i;
    }

    static int staticMethod(int i)
    {
        return --i;
    }
}

class Y extends X
{
    public Y()
    {
        System.out.println(staticMethod(i));
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Y y = new Y();
    }
}
```

View Answer

Answer :
101010

23) What happens if both super class and sub class have a field with same name?

View Answer

Answer :
Super class field will be hidden in the sub class.

24) Does the below program execute successfully? If yes, what will be the output?

```
class A
{
    void A()
    {
        System.out.println(1);
    }
}

class B extends A
{
    void B()
    {
        A();
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        new B().B();
    }
}
```

View Answer

Answer :
Yes, but with a warning that method has a constructor name. Output will be
1

25) How do you prevent a field or a method of any class from inheriting to sub classes?

View Answer

Answer :
By declaring that particular field or method as private.

26) What will be the output of the below program?

```
class A
{
    int i = 1212;
}
```

```
class B extends A
{
    A a;

    public B(A a)
    {
        this.a = a;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        A a = new A();

        B b = new B(a);

        System.out.println(a.i);

        System.out.println(b.i);

        System.out.println(b.a.i);

        b.a.i = 2121;

        System.out.println("--------");

        System.out.println(a.i);

        System.out.println(b.i);
    }
}
```

View Answer

Answer :
1212
1212
1212

2121
1212

27) Does java support multiple inheritance?

View Answer

Answer :
Yes, but only through interfaces. Classes can implement more than one interfaces but can not extend more than one class.

28) What will be the output of this program?

```
class ClassOne
{
     static int i, j = 191919;

     {
         --i;
     }

     {
         j++;
     }
}

public class ClassTwo extends ClassOne
{
    static
    {
        i++;
    }

    static
    {
        --j;
    }

    public static void main(String[] args)
    {
        System.out.println(i);

        System.out.println(j);
    }
}
```

View Answer

Answer :
1
191918

29) Does the fields with default visibility inherited to sub classes outside the package?

View Answer

Answer :
No. Fields with default access modifier are inherited to sub classes within the package.

30) Constructors are also inherited to sub classes. True or false?

View Answer

Answer :
false. Constructors, SIB and IIB are not inherited to sub classes. But, they are executed while instantiating a subclass.

31) What will be the output of the below program?

```
class A
{
    int[] a = new int[5];

    {
        a[0] = 10;
    }
}

public class MainClass extends A
{
    {
        a = new int[5];
    }

    {
        System.out.println(a[0]);
    }

    public static void main(String[] args)
    {
        MainClass main = new MainClass();
    }
}
```

View Answer

32) What happens if both super class and sub class have a field with same name?

View Answer

Answer :
Super class field will be hidden in the sub class.

33) What will be the output of the below program?

```
class A
{
    int methodOfA(int i)
    {
        i /= 10;

        return i;
    }
}

class B extends A
{
    int methodOfB(int i)
    {
        i *= 20;

        return methodOfA(i);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        B b = new B();

        System.out.println(b.methodOfB(100));
    }
}
```

View Answer

Answer :
200

34) What will be the outcome of the following program?

```
class A
{
    static int i;

    static
    {
        i++;
    }

    {
        ++i;
    }
}

class B extends A
{
    static
    {
        --i;
    }

    {
        i--;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        System.out.println(new B().i);
    }
}
```

View Answer

35) Can a class be extended by more than one classes?

View Answer

Answer :
Yes, A class in java can be extended by multiple classes.

36) Does the below program written correctly? If yes, what will be the output?

```
public class MainClass
{
    public MainClass(int i, int j)
    {
        System.out.println(method(i, j));
    }

    int method(int i, int j)
    {
        return i++ + ++j;
    }

    public static void main(String[] args)
    {
        MainClass main = new MainClass(10, 20);
    }
}
```

View Answer

Answer :

Yes, class is written correctly. Output will be 31.

37) Does the below code prints "Hi…." on the console? If yes, how many times?

```
class X
{
    static
    {
        Y.methodOfY();
    }
}

class Y extends X
{
    static void methodOfY()
    {
        System.out.println("Hi....");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Y.methodOfY();
    }
}
```

View Answer

Answer :
Above code prints 2 times "Hi…." on the console.

38) What value the fields 'i' and 'j' will hold when you instantiate 'ClassTwo' in the below code?

```
class ClassOne
{
    static int i = 111;

    int j = 222;

    {
        i = i++ - ++j;
    }
}

class ClassTwo extends ClassOne
{
    {
        j = i-- + --j;
    }
}
```

View Answer

39) When you instantiate a sub class, super class constructor will be also executed. True or False?

View Answer

Answer :

True

40) Does the below code written correctly? If yes, what will be the output?

```
class One
{
    int x = 2121;
}

class Two
{
    int x = 1212;

    {
        System.out.println(x);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Two two = new Two();
    }
}
```

View Answer

Answer :
Yes, above code is written correctly. Output will be 1212.

Java Practice Questions On Classes And Objects

1) Which of the following is true about Static Initialization Block?

A. We can use only static members of a class inside the Static Initialization Block.
B. Static Initialization Blocks are mainly used to initialize static fields of a class.
C. Static Initialization Block is the first block to be executed after class is loaded in the memory.
D. All of the above.

View Answer

Answer :
D. All of the above.Static Initialization Blocks are the first blocks to be executed after class is loaded in the memory. These blocks are mainly used to initialize static fields of a class. Inside a static initialization block, we can use only static members of a class.

2) Where the static initialization blocks are stored in the memory?

View Answer

Answer :
No Where. Static Initialization Blocks are not at all stored in the memory. They just come to stack, execute their task and leave the memory.

3) What will be the output of this program?

```
class A
{
    static int i;

    static
    {
        System.out.println(1);

        i = 100;
    }
}

public class StaticInitializationBlock
{
    static
    {
        System.out.println(2);
    }

    public static void main(String[] args)
    {
        System.out.println(3);

        System.out.println(A.i);
    }
}
```

View Answer

Answer :
2
3
1
100

4) What happens when you compile the below class?

```
class A
{
    int i;

    static
    {
        System.out.println(i);
    }
}
```

View Answer

Answer :
It gives compile time error (Line 7). Because, you can’t refer a non static field inside a static initialization block.

5) Is the below code written correctly?

```
class A
{
    static
    {
        static
        {
            System.out.println(1);
        }
    }
}
```

View Answer

Answer :
No. Static Initialization Blocks can not be nested.

6) How many static initialization blocks are there in the below Class A?

```
class A
{
    static int a = 50;

    static
    {
        a = 50;
    }

    static
    {
        a = 50;
    }
}
```

View Answer

Answer :
Three.

```
class A
{
    static int a = 50;          // ----- (1)

    static
    {
        a = 50;          // ----- (2)
    }

    static
    {
        a = 50;          // ----- (3)
    }
}
```

7) What will be the outcome of the following program?

```
public class A
{
    static
    {
        System.out.println(1);
    }

    static
    {
        System.out.println(2);
    }

    static
    {
        System.out.println(3);
    }

    public static void main(String[] args)
    {
        A a;
    }
}
```

View Answer

Answer :
1
2
3

8) What will be the output of this program?

```
class A
{
    static int first;

    static String second;

    static
    {
        System.out.println(1);

        first = 100;
    }

    static
    {
        System.out.println(2);

        second = "SECOND";
    }
}

public class StaticInitializationBlock
{
    static
    {
        System.out.println(3);
    }

    public static void main(String[] args)
    {
        System.out.println(4);

        System.out.println(A.first);

        System.out.println(A.second);
    }
}
```

View Answer

Answer :
3
4
1
2
100
SECOND

9) What will be the output of the below program?

```
class A
{
    static int i;

    static
    {
        i = 100;

        System.out.println(1);
    }

    static void staticMethod()
    {
        System.out.println(i);

        System.out.println(2);
    }
}

public class B
{
    static
    {
        System.out.println(3);
    }

    public static void main(String[] args)
    {
        System.out.println(4);

        System.out.println(A.i);

        A.staticMethod();
    }
}
```

View Answer

Answer :
3
4
1
100
100
2

10) What is the difference between SIB and IIB?

View Answer

Answer :
SIB – Static Initialization Block, is the first block to be executed after class is loaded in the memory. This block is mainly used to initialize static members of a class.IIB – Instance Initialization Block, is executed while instantiating a class. This block is executed each time you create an object to the class. This block is mainly used to initialize instance members of a class.

1) Below class ABC doesn’t have even a single abstract method, but it has been declared as abstract. Is it correct?

```
abstract class ABC
{
    void firstMethod()
    {
        System.out.println("First Method");
    }

    void secondMethod()
    {
        System.out.println("Second Method");
    }
}
```

View Answer

Answer :
Yes, it is correct. abstract classes may or may not have abstract methods.

2) Why the below class is showing compilation error?

```
abstract class AbstractClass
{
    abstract void abstractMethod()
    {
        System.out.println("First Method");
    }
}
```

View Answer

Answer :
Because, abstract methods must not have a body.

3) Which class is instantiable? Class A or Class B?

```
abstract class A
{

}

class B extends A
{

}
```

View Answer

Answer :
Class B.

4) Below code snippet is showing compilation error? Can you suggest the corrections?

```
abstract class A
{
    abstract int add(int a, int b);
}

class B extends A
{

}
```

View Answer

Answer :
Class B must implement inherited abstract method A.add() or else it must be declared as abstract.

5) Is the following program written correctly? If yes, what value “result” variable will hold if you run the program?

```
abstract class Calculate
{
    abstract int add(int a, int b);
}

public class MainClass
```

```
{
    public static void main(String[] args)
    {
        int result = new Calculate()
        {
            @Override
            int add(int a, int b)
            {
                return a+b;
            }
        }.add(11010, 022011);
    }
}
```

View Answer

Answer :
Yes, program is written correctly. result = 20235.

6) Can we write explicit constructors in an abstract class?

View Answer

Answer :
Yes. abstract classes can have any number of constructors.

7) Can you identify the error in the below code?

```
abstract class AbstractClass
{
    private abstract int abstractMethod();
}
```

View Answer

Answer :
abstract methods can’t be private.

8) Can we declare protected methods in an interface?

View Answer

Answer :
No. only public methods are allowed in an interface.

9) What will be the output of the following program?

```
abstract class A
{
    abstract void firstMethod();

    void secondMethod()
    {
        System.out.println("SECOND");

        firstMethod();
    }
}

abstract class B extends A
{
    @Override
    void firstMethod()
    {
        System.out.println("FIRST");

        thirdMethod();
    }

    abstract void thirdMethod();
}

class C extends B
{
    @Override
    void thirdMethod()
    {
        System.out.println("THIRD");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();

        c.firstMethod();

        c.secondMethod();

        c.thirdMethod();
    }
}
```

View Answer

Answer :
FIRST
THIRD
SECOND
FIRST
THIRD
THIRD

10) What will be the output of the below program?

```
abstract class X
{
    public X()
    {
        System.out.println("ONE");
    }

    abstract void abstractMethod();
}

class Y extends X
{
    public Y()
    {
        System.out.println("TWO");
    }

    @Override
    void abstractMethod()
    {
        System.out.println("THREE");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        X x = new Y();

        x.abstractMethod();
    }
}
```

View Answer

Answer :
ONE
TWO
THREE

1) For every interface written in a java file, .class file will be generated after compilation? True or False?

View Answer

Answer :
True. For every interface written in a java file, .class file will be generated after compilation.

2) Can you identify the error in the below code?

```
interface A
{
    private int i;
}
```

View Answer

Answer :
Illegal modifier for field i. Only public, static and final are allowed.

3) What will be the output of the following program?

```
interface A
{
    void myMethod();
}

class B
```

```
{
    public void myMethod()
    {
        System.out.println("My Method");
    }
}

class C extends B implements A
{

}

class MainClass
{
    public static void main(String[] args)
    {
        A a = new C();

        a.myMethod();
    }
}
```

View Answer

Answer :
My Method

4) Can a class implement more than one interfaces?

View Answer

Answer :
Yes, a class can implement more than one interfaces.

5) Why the below code is showing compile time error?

```
interface X
{
    void methodX();
}

class Y implements X
{
    void methodX()
    {
        System.out.println("Method X");
    }
}
```

View Answer

Answer :
Interface methods must be implemented as public. Because, interface methods are public by default and you should not reduce the visibility of any methods while overriding.

6) Does below code compile successfully? If not, why?

```
interface A
{
    int i = 111;
}

class B implements A
{
    void methodB()
    {
        i = 222;
    }
}
```

View Answer

Answer :
No, because interface fields are static and final by default and you can’t change their value once they are initialized. In the above code, methodB() is changing value of interface field A.i. It shows compile time error.

7) Is the following code written correctly?

```
class A
{
    //Class A
}

interface B extends A
{
    //Interface B extending Class A
}
```

View Answer

Answer :
No. An interface can extend another interface not the class.

8) What will be the output of the following program?

```
interface P
{
    String p = "PPPP";

    String methodP();
}

interface Q extends P
{
    String q = "QQQQ";

    String methodQ();
}
```

```
class R implements P, Q
{
    public String methodP()
    {
        return q+p;
    }

    public String methodQ()
    {
        return p+q;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        R r = new R();

        System.out.println(r.methodP());

        System.out.println(r.methodQ());
    }
}
```

View Answer

Answer :
QQQQPPPP
PPPPQQQQ

9) Can interfaces have constructors?

View Answer

Answer :
No. Interfaces can't have constructors.

10) Is the below program written correctly? If yes, what will be the output?

```
class A implements B
{
    public int methodB(int i)
    {
        return i =+ i * i;
    }
}

interface B
{
    int methodB(int i);
}

public class MainClass
{
    public static void main(String[] args)
    {
        B b = new A();

        System.out.println(b.methodB(2));
    }
}
```

View Answer

Answer :
Yes, program is written correctly. Output will be,
4

11) Can you find out the errors in the following code?

```
interface A
{
    {
        System.out.println("Interface A");
    }

    static
    {
        System.out.println("Interface A");
    }
}
```

View Answer

Answer :
Interfaces can’t have initializers.

12) How do you access interface field ‘i’ in the below code?

```
class P
{
    interface Q
    {
        int i = 111;
    }
}
```

View Answer

Answer :
P.Q.i

40 Java Practice Questions On Method Overloading And Overriding :

1) What will be the output of the following program?

```
class A
{

}

class B extends A
{

}

class C extends B
{

}

public class MainClass
{
    static void overloadedMethod(A a)
    {
        System.out.println("ONE");
    }

    static void overloadedMethod(B b)
    {
        System.out.println("TWO");
    }

    static void overloadedMethod(Object obj)
    {
        System.out.println("THREE");
    }

    public static void main(String[] args)
    {
        C c = new C();

        overloadedMethod(c);
    }
}
```

View Answer

Answer :
TWO

2) In a class, one method has two overloaded forms. One form is defined as static and another form is defined as non-static. Is that method properly overloaded?

View Answer

Answer :
Yes. Compiler checks only method signature to verify whether a particular method is properly overloaded or not. It doesn’t check static or non-static feature of the method.

3) In the below class, is ‘method’ overloaded or duplicated?

```java
public class MainClass
{
    void method(int ... a)
    {
        System.out.println(1);
    }

    void method(int[] a)
    {
        System.out.println(2);
    }
}
```

View Answer

Answer :
Duplicated. Because, var args (int … a) are nothing but the arrays. So here, (int … a) and (int[] a) are the same.

4) Method signature consists of
a) Method Name, Return Type and Number Of Arguments
b) Access Modifier, Method Name and Types Of Arguments
c) Method Name, Number Of Arguments, Types Of Arguments and Order Of Arguments
d) Return Type, Access Modifier and Order Of Arguments

View Answer

5) In the below Class X, is ‘method’ properly overloaded?

```java
class X
{
    int method(int i, int d)
    {
        return i+d;
    }

    static int method(int i, double d)
    {
        return (int)(i+d);
    }

    double method(double i, int d)
    {
        return i+d;
    }

    static double method(double i, double d)
    {
        return i+d;
    }
}
```

View Answer

Answer :
Yes.

6) What will be the output of the following program?

```
class X
{
    void method(int a)
    {
        System.out.println("ONE");
    }

    void method(double d)
    {
        System.out.println("TWO");
    }
}

class Y extends X
{
    @Override
    void method(double d)
    {
        System.out.println("THREE");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        new Y().method(100);
    }
}
```

View Answer

Answer :
ONE

7) What will be the outcome of the below program?

```
public class MainClass
{
    double overloadedMethod(double d)
    {
        return d *= d;
    }

    int overloadedMethod(int i)
    {
        return overloadedMethod(i *= i);
    }

    float overloadedMethod(float f)
    {
        return overloadedMethod(f *= f);
    }
```

```java

    public static void main(String[] args)
    {
        MainClass main = new MainClass();

        System.out.println(main.overloadedMethod(100));
    }
}
```

View Answer

Answer :
It will throw java.lang.StackOverflowError at run time. Because, overloadedMethod(int) keeps calling itself.

8) Method Overriding shows static polymorphism. True or false?

View Answer

Answer :
False. Method Overriding shows dynamic polymorphism.

9) In a class, One method has 4 overloaded forms. All have different access modifiers (private, default, protected and public). Is that method properly overloaded?

View Answer

Answer :
Yes. That method is properly overloaded.

Also Read : Increment And Decrement Operators Quiz

10) What will be the outcome of the following program?

```java
class X
{
    void calculate(int a, int b)
    {
        System.out.println("Class X");
    }
}

class Y extends X
{
    @Override
    void calculate(int a, int b)
    {
        System.out.println("Class Y");
    }
}

class Z extends Y
{
    @Override
    void calculate(int a, int b)
    {
        System.out.println("Class Z");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        X x = new Y();

        x.calculate(10, 20);

        Y y = (Y) x;
36
37          y.calculate(50, 100);
38
39          Z z = (Z) y;
40
41          z.calculate(100, 200);
42      }
43  }
```

View Answer

Answer :
Line 39 (Z z = (Z) y) will throw java.lang.ClassCastException at run time. Because, Y cannot be cast to Z.

11) Will you find out the error in the below code?

```
class X
{
    static void methodOfX()
    {
        System.out.println("Class X");
    }
}

class Y extends X
{
    @Override
    static void methodOfX()
    {
        System.out.println("Class X");
    }
}
```

View Answer

Answer :
Can’t override static methods.

12) What possible types a ‘superClassMethod()’ of below ‘SuperClass’ can return when it is overridden in the sub class?

```
class SuperClass
{
    Object superClassMethod()
    {
        return new Object();
    }
}
```

View Answer

Answer :
Any type. Because Object class is a super class for all the classes in Java.

13) Can we override protected method as private?

View Answer

Answer :
No. While overriding, visibility of a method can be increased but can not be reduced.

14) What will be the output of this program?

```
class SuperClass
{
    void superClassMethod(Number n)
    {
        System.out.println("From Super Class");
    }
}

class SubClass extends SuperClass
{
    void superClassMethod(Double d)
    {
        System.out.println("From Sub Class");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        SubClass sub = new SubClass();

        sub.superClassMethod(123321);
    }
}
```

View Answer

Answer :
From Super Class

15) What actually polymorphism means in Java?

View Answer

Answer :
Polymorphism in java refers to any entity whether it is an operator or a constructor or any method which takes many forms or can be used for multiple tasks, either while compiling or running a java program.

16) Does Java support operator overloading?

View Answer

Answer :
Java doesn’t support operator overloading. (exception being ‘+’ symbol which is used for both addition of two numbers as well as for concatenation of two strings.)

17) What will be the output of the below program?

```
class A
{
    public A(int i)
    {
        System.out.println(1);
    }

    public A()
    {
        this(10);

        System.out.println(2);
    }

    void A()
    {
        A(10);

        System.out.println(3);
    }

    void A(int i)
    {
        System.out.println(4);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        new A().A();
    }
}
```

View Answer

Answer :
1
2
4
3

18) Why method overriding is called late binding or dynamic binding?

View Answer

Answer :
Because object used for binding will be known only at run time.

19) What will be the output of the following program?

```
public class MainClass
{
    static void method(Integer i)
    {
        System.out.println(1);
    }

    static void method(Double d)
    {
        System.out.println(2);
    }

    static void method(Number n)
    {
        System.out.println(4);
    }

    static void method(Object o)
    {
        System.out.println(5);
    }

    public static void main(String[] args)
    {
        method((short)12);
    }
}
```

View Answer

Answer :
4

20) How do compiler differentiate overloaded methods from duplicate methods?

View Answer

Answer :
Compiler uses method signature to check whether the method is overloaded or duplicated. Duplicate methods will have same method signatures i.e same name, same number of arguments and same types of arguments. Overloaded methods will have same name but differ in number of arguments or in types of arguments.

Also Read : Java Strings Quiz

21) final methods can be overridden but can’t be overloaded? True or False?

View Answer

Answer :
False. final methods can be overloaded but can’t be overridden.

22) Does the below program shows polymorphism or not?

```
class A
{

}

class B extends A
{

}

class C extends B
{

}

public class MainClass
{
    public static void main(String[] args)
    {
        A a = new A();

        a = new B();

        a = new C();
    }
}
```

View Answer

Answer :
Yes. Class-A type reference variable is referring to Class-A type object, Class-B type object and Class-C type object. This shows the polymorphism.

23) What will be the output of the following program?

```java
class X
{
    int method(int i)
    {
        return i *= i;
    }
}

class Y extends X
{
    double method(double d)
    {
        return d /= d;
    }
}

class Z extends Y
{
    float method(float f)
    {
        return f += f;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Z z = new Z();

        System.out.println(z.method(21.12));
    }
}
```

View Answer

Answer :
1.0

24) What will be the output of the following program?

```java
class ClassOne
{
    void method(String s1)
    {
        method(s1, s1+s1);
    }

    void method(String s1, String s2)
    {
        method(s1, s2, s1+s2);
    }

    void method(String s1, String s2, String s3)
    {
        System.out.println(s1+s2+s3);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        ClassOne one = new ClassOne();

        one.method("JAVA");
    }
}
```

View Answer

Answer :
JAVAJAVAJAVAJAVAJAVAJAVA

25) Constructor overloading is also one form of the polymorphism. Yes or No?

View Answer

Answer :
Yes. Constructor overloading is also one form of polymorphism.

26) Is the following program written correctly? If yes, what will be the output?

```
class A
{
    public A(int i)
    {
        System.out.println(myMethod(i));
    }

    int myMethod(int i)
    {
        return ++i + --i;
    }
}

class B extends A
{
    public B(int i, int j)
    {
        super(i*j);

        System.out.println(myMethod(i, j));
    }

    int myMethod(int i, int j)
    {
        return myMethod(i*j);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        B b = new B(12, 21);
    }
}
```

View Answer

Answer :
505
505

27) What will be the output of the below program?

```
class A
{
    void myMethod(Object o, Double D)
    {
        System.out.println(1);
    }

    void myMethod(Integer I, Number N)
    {
        System.out.println(2);
    }
}
```

```

class B extends A
{
    void myMethod(Float F, Double D)
    {
        System.out.println(3);
    }

    void myMethod(Double D, Integer I)
    {
        System.out.println(4);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        B b = new B();

        b.myMethod(11.11, 0000);

        b.myMethod(8778, 3223);

        b.myMethod(2.3*1.2, 4.1*1.4);

        b.myMethod((float)23.56, 21.45);
    }
}
```

View Answer

Answer :
4
2
1
3

28) In the below example, Class B extends Class A. Which method of Class A is not properly overridden in class B?

```java
class A
{
    void methodOne(Double D)
    {

    }

    int methodTwo(Integer I)
    {
        return I;
    }
}

class B extends A
{
    @Override
    void methodOne(double d)
    {

    }

    @Override
    int methodTwo(Integer I)
    {
        return (int)1.1;
    }
}
```

View Answer

Answer :
methodOne() is not properly overridden. Because, arguments are not compatible.

29) Can we access super class version of overridden method in the sub class. If yes, how?

View Answer

Answer :
Yes. We can access super class version of overridden method in the sub class using **super** keyword.

Also Read : Java Inheritance Quiz

30) What is the difference between static binding and dynamic binding?

View Answer

Answer :
Click here to see the differences between static binding and dynamic binding.

31) What will be the output of the following program?

```
class A
{
    static void methodOne()
    {
        System.out.println("AAA");
    }
}

class B extends A
{
    static void methodOne()
    {
        System.out.println("BBB");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        A a = new B();

        a.methodOne();
    }
}
```

View Answer

Answer :
AAA

32) In the below class A, 'myMethod()' has three different forms. All are throwing different exceptions, but have same signature. Is it OK?

```
class A
{
    void myMethod() throws IOException
    {
        System.out.println("ONE");
    }

    void myMethod() throws NumberFormatException
    {
        System.out.println("TWO");
    }

    void myMethod() throws ArrayIndexOutOfBoundsException
    {
        System.out.println("THREE");
    }
}
```

View Answer

Answer :
It is not Ok. You will get duplicate method error.

33) Can you identify the error in below code snippet?

```
class A
{
    void myMethod()
    {
        System.out.println("Super Class");
    }
}

class B extends A
{
    @Override
    void myMethod() throws SQLException
    {
        System.out.println("Sub Class");
    }
}
```

View Answer

Answer :
SQLException is not compatible with throws clause of super class method. If super class method doesn’t have throws clause, then it can be overridden with only unchecked type of exceptions. SQLException is not an unchecked type of exception.

34) Can we remove throws clause of a method while overriding it?

View Answer

Answer :
Yes, we can remove throws clause of a method while overriding it.

35) What will be the outcome of the following program?

```
class ABC
{
    void methodABC()
    {
        new XYZ().methodXYZ();
    }
}

class XYZ extends ABC
{
    void methodXYZ()
    {
        methodABC();
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        new ABC().methodABC();
    }
}
```

View Answer

Answer :
You will get StackOverflowError.

36) What will be the output of the below program?

```
class ABC
{
    void methodABC()
    {
        System.out.println(111);
    }

    void methodABC(int i)
    {
        System.out.println(222);
    }
}

class XYZ extends ABC
{
    @Override
    void methodABC(int i)
    {
        System.out.println(333);
    }

    @Override
    void methodABC()
    {
        System.out.println(444);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        ABC abc = new XYZ();

        abc.methodABC(10);

        abc.methodABC();
    }
}
```

View Answer

Answer :
333
444

37) What are the possible access modifiers a protected method can have if it is overridden in the sub class?

View Answer

Answer :
protected or public.

38) Can you list out the differences between method overloading and method overriding?

View Answer

Answer :
Click here to see the differences between method overloading and method overriding.

39) In the below example, is “methodOfX()” correctly overridden in the sub classes of Class X?

```
class X
{
    void methodOfX()
    {
        System.out.println("Class X");
    }
}

class Y extends X
{
    @Override
    protected void methodOfX()
    {
        System.out.println("Class Y");
    }
}

class Z extends Y
{
    @Override
    public void methodOfX()
    {
        System.out.println("Class Z");
    }
}
```

View Answer

Answer :
Yes.

40) What will be the output of the following program?

```
class ABC
{
    String myMethod(String s)
    {
        return s+s;
    }
}

class PQR extends ABC
{
    String myMethod(String s, double d)
    {
        return myMethod(s+d);
    }
}

class XYZ extends PQR
{
    String myMethod(String s, double d, int i)
    {
        return myMethod(s, d+i);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        XYZ xyz = new XYZ();

        System.out.println(xyz.myMethod("JAVA", 23.23, 111));
    }
}
```

View Answer

Answer :
JAVA134.23JAVA134.23

https://www.youtube.com/@codewitharrays

https://www.instagram.com/codewitharrays/

https://t.me/codewitharrays Group Link: https://t.me/ccee2025notes

+91 8007592194 +91 9284926333

codewitharrays@gmail.com

https://codewitharrays.in/project