

# *Practice Problems*

Q1. What will be the output of the 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);
    }
}
```

Q2. What is the error in the following code:

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

class Y
{
    //Class Y Members
}

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

Q3. What will be the output of the 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();
    }
}

```

**Q4. what is will be the output of the 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();
    }
}

```

```
    }  
}
```

Q5. The following code has compilation error. What is the reason for the error?

```
class X  
{  
    public X(int i)  
    {  
        System.out.println(1);  
    }  
}  
  
class Y extends X  
{  
    public Y()  
    {  
        System.out.println(2);  
    }  
}  
  
Class C  
{  
    public static void main(String args[])  
    {  
        Z=new Y()  
    }  
}
```

Q6. What is the error in the following code:

```
public class A  
{  
    public A(int i)  
    {  
    }  
}  
  
class B extends A  
{  
  
}
```

Q7. What is the error in the code?

```
public class A  
{  
    public A()  
    {  
        super();  
  
        this(10);  
    }  
}
```

```

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

```

**Q8. What is the error in the code?**

```

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();
    }
}

```

**Q9. You know that compiler will implicitly keep `super()` calling statement as a first statement in every constructor. What happens if we write `this()` as a first statement in our constructor?**

Q10. Can a class extend itself?

Q11. Does Java support multiple inheritance?

Q12. what will be the output of the 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)); 200
    }
}
```

Q13. Is the following fragment of code correct? justify

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

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

Q14. What is the error in the code fragment?

No abstract before void

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

Q14. What is the error in the code?

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

class B extends A
{
    Write implementation details of add
}
```

Q15. Is the following code correct? In case it is correct, what is the output

```
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);
    }
}
```

Output-> Nothing

Q16. What is the output of the 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();
    }
}

```

FIRST  
THIRD  
SECOND  
FIRST  
THIRD  
THIRD

Q17. what is wrong in the code?

```

interface A
{
    private int i;    public, static, final only allowed
}

```

Q18. Does the code compile? Why or Why not?

```

interface X
{
    void methodX();
}

```

```

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

```

Q19. Does the code compile? Why or Why not?

```

interface A
{
    int i = 111;
}

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

```

The final field A.i cannot be assigned

Q20. 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?

Q21. In the below class, is 'method' overloaded or duplicated?

```

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

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

```

Overloaded

Q22. Discuss the trace of execution of the program?

```

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

```

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);
        }
    }

```

Q23. Trace the order of print statement of the program?

```

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

    public A()
    {
        System.out.println(2);
    }

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

        A a = new A();
    }
}

```

3  
1  
2

Q24. what is the error in the code?

```

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

    int i = 10;
}

```

Cannot reference a field before it is defined

Q25. Trace the output of the program

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        System.out.println(1);

        try
        {
            System.out.println(2);

            int i = Integer.parseInt("ABC");

            System.out.println(3);
        }
        catch (Exception e)
        {
            System.out.println(4);
        }

        finally
        {
            System.out.println(5);
        }

        System.out.println(6);
    }
}
```