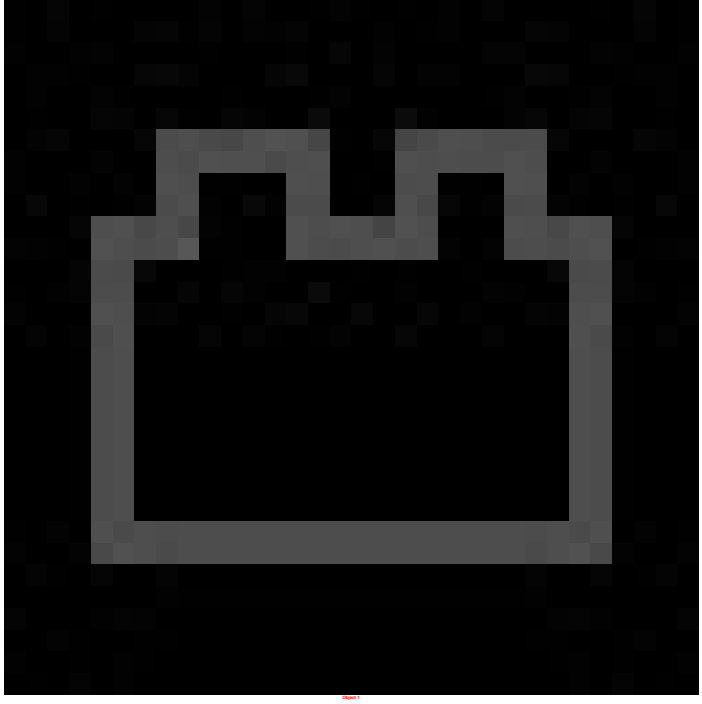

DSA-Quiz-Feb’2020

1. Find output of the following code.

```
class Helper
{
	private int data;
	private Helper()
	{
		data = 5;
	}
}
public class Test
{
	public static void main(String[] args)
	{
		Helper help = new Helper();
		System.out.println(help.data);
	}
}
```

a) Compilation error
b) 5
c) Runtime error
d) None of these
Ans: (a)

2. Find output of the following code.

```
class Temp
{
	private Temp(int data)
	{
		System.out.printf(" Constructor called ");
	}
	protected static Temp create(int data)
```

```
	{
		Temp obj = new Temp(data);
		return obj;
	}
	public void myMethod()
	{
		System.out.printf(" Method called ");
	}
}

public class Test
{
	public static void main(String[] args)
	{
		Temp obj = Temp.create(20);
		obj.myMethod();
	}
}
```

a) Constructor called Method called
b) Compilation error
c) Runtime error
d) None of the above
Ans: (a)

3. Find output of the following code.

```
class Base
{
	public static String s = " Super Class ";
	public Base()
	{
		System.out.printf("1");
	}
}
class Derived extends Base
{
	public Derived()
	{
		System.out.printf("2");
		super();
	}
}
public class Test
{
	public static void main(String[] args)
	{
		Derived obj = new Derived();
		System.out.printf(s);
	}
}
```

a) 21 Super Class
b) Super Class 21

c) Compilation error
d) 12 Super Class
Ans. (c)

4. Find output of the following code.

```
class Derived
{
	protected final void getDetails()
	{
		System.out.println("Derived class");
	}
}

class Test extends Derived
{
	protected final void getDetails()
	{
		System.out.println("Test class");
	}
}
public class Tester
{
	public static void main(String[] args)
	{
		Derived obj = new Derived();
		obj.getDetails();
	}
}
```

a) Derived class
b) Test class
c) Runtime error
d) Compilation error

Ans. (d)

5. Find output of the following code.

```
class Derived
{
	public void getDetails(String temp)
	{
		System.out.println("Derived class " + temp);
	}
}
class Test extends Derived
{
	public int getDetails(String temp)
	{
```

```
            System.out.println("Test class " + temp);
            return 0;
        }
}
public class Tester
{
        public static void main(String[] args)
        {
            Test obj = new Test();
            obj.getDetails("GFG");
        }
}
```

a) Derived class GFG
b) Test class GFG
c) Compilation error
d) Runtime error

Ans. (c)

6. Find output of the following code.

```
class Derived
{
        public void getDetails()
        {
            System.out.printf("Derived class ");
        }
}
class Test extends Derived
{
        public void getDetails()
        {
            System.out.printf("Test class ");
            super.getDetails();
        }
}
public class Tester
{
        public static void main(String[] args)
```

```
    {
        Derived obj = new Test();
        obj.getDetails();
    }
}
```

a) Test class Derived class
b) Derived class Test class
c) Compilation error
d) Runtime error

Ans. (a)

7. Find output of the following code.

```
class Derived
{
    public void getDetails()
    {
        System.out.println("Derived class");
    }
}
public class Test extends Derived
{
    protected void getDetails()
    {
        System.out.println("Test class");
    }
    public static void main(String[] args)
    {
        Derived obj = new Test(); // line xyz
        obj.getDetails();
    }
}
```

a) Test class
b) Compilation error due to line xyz
c) Derived class
d) Compilation error due to access modifier

Ans: (d)

8. Find output of the following code.

```
public class Test
{
	public int getData() //getdata() 1
	{
		return 0;
	}
	public long getData() //getdata 2
	{
		return 1;
	}
	public static void main(String[] args)
	{
		Test obj = new Test();
		System.out.println(obj.getData());
	}
}
```

a) 1
b) 0
c) Runtime error
d) Compilation error

Ans. (d)

9. Find output of the following code.

```
class B
{
	public static String sing()
	{
		return "la";
	}
}
class A extends B
{
	public static String sing()
	{
		return "fa";
```

```
    }
}
public class Test
{
    public static void main(String[] args)
    {
        A a = new A();
        B b = new A();
    System.out.println(a.sing() + " " + b.sing());
    }
}
```

Output: fa la

10. Find output of the following code.

```
class Building
{
    Building()
    {
        System.out.println("Buiding");
    }
    Building(String name)
    {
        this();
        System.out.println("Building: String Constructor" + name);
    }
}
class House extends Building
{
    House()
    {
        System.out.println("House ");
    }
    House(String name)
    {
        this();
```

```
            System.out.println("House: String Constructor" + name);
        }
}
public class Test
{
        public static void main(String[] args)
        {
            new House("  OK");
        }
}
```

Output:

```
Buiding
House
House: String Constructor  OK
```

11. Which of the following is FALSE about abstract classes in Java

(a) If we derive an abstract class and do not implement all the abstract methods, then the derived class should also be marked as abstract using ‘abstract’ keyword
(b) Abstract classes can have constructors
(c)A class can be made abstract without any abstract method
(d)A class can inherit from multiple abstract classes.

Answer: (d)

12. Which of the following is true about interfaces in java.

1) An interface can contain following type of members.
   public, static, final fields (i.e., constants)
   public and abstract methods

2) An instance of interface can be created.

3) A class can implement multiple interfaces.

4) Many classes can implement the same interface.

(A) 1, 3 and 4
(B) 1, 2 and 4
(C) 2, 3 and 4
(D) 1, 2, 3 and 4

Answer: (A)

13. Find output of the following code.

```
abstract class demo
{
	public int a;
	demo()
	{
		a = 10;
	}
	abstract public void set();
	abstract final public void get();
}
class Test extends demo
{
	public void set(int a)
	{
		this.a = a;
	}
	final public void get()
	{
		System.out.println("a = " + a);
	}
}
public class Tester
{
	public static void main(String[] args)
	{
		Test obj = new Test();
		obj.set(20);
		obj.get();
	}
}
```

(A) a = 10
(B) a = 20
(C) Compilation error
(D) None of these

Answer: (C)

14. Find output of the following code.

```
package main;
class T {
	int t = 20;
}
public class Main {
	public static void main(String args[]) {
		T t1 = new T();
		System.out.println(t1.t);
	}
}
```

(A) 20
(B) 0
(C) Compiler Error
(D) None of these

Answer: (A)

15. Find output of the following code.

```
class T {
int t = 20;
T() {
	t = 40;
}
}
public class Main {
	public static void main(String args[]) {
		T t1 = new T();
		System.out.println(t1.t);
	}
}
```

(A) 20
(B) 40
(C) Compiler Error
(D) None of these

Answer: (B)

16. Which of the following is/are true about constructors in Java?

1) Constructor name should be same as class name.
2) If you don't define a constructor for a class,

a default parameterless constructor is automatically created by the compiler.
3) The default constructor calls super() and initializes all instance variables to default value like 0, null.
4) If we want to parent class constructor, it must be called in first line of constructor.

(A) 1
(B) 1, 2
(C) 1, 2 and 3
(D) 1, 2, 3 and 4

Answer: (D)

17. Find output of the following code.

```
final class Complex {
        private double re, im;
        public Complex(double re, double im) {
                this.re = re;
                this.im = im;
        }
        Complex(Complex c)
        {
        System.out.println("Copy constructor called");
        re = c.re;
        im = c.im;
        }
        public String toString() {
                return "(" + re + " + " + im + "i)";
        }
}
public class Main {
        public static void main(String[] args) {
                Complex c1 = new Complex(10, 15);
                Complex c2 = new Complex(c1);
                Complex c3 = c1;
                System.out.println(c2);
        }
}
```

(A)

Copy constructor called
(10.0 + 15.0i)

(B)

Copy constructor called
(0.0 + 0.0i)

(C)

(10.0 + 15.0i)

(D)

(0.0 + 0.0i)

Answer: (A)

18. Find output of the following code.

```
public class Main
{
	public static void gfg(String s)
	{
		System.out.println("String");
	}
	public static void gfg(Object o)
	{
		System.out.println("Object");
	}

	public static void main(String args[])
	{
		gfg(null);
	}
}
```

Output: String

19. Find output of the following code.

```
class First
```

```
{
        public First() { System.out.println("a"); }
}

class Second extends First
{
        public Second() { System.out.println("b"); }
}

class Third extends Second
{
        public Third() { System.out.println("c"); }
}

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

Output:

a

b

c

20. Find output of the following code.

```
class First
{
        int i = 10;

        public First(int j)
        {
                System.out.println(i);
                this.i = j * 10;
```

```
    }
}

class Second extends First
{
    public Second(int j)
    {
        super(j);
        System.out.println(i);
        this.i = j * 20;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Second n = new Second(20);
        System.out.println(n.i);
    }
}
```

Output:

10

200

400

21. Which of the following is FALSE about finally block?

a) For each try block, there can be only 1 finally block.
b) finally block will not be executed if program exits by calling System.exit();
c) If an exception is not handled in the catch statement, before terminating the program, JVM executes the finally block
d) finally block contains important code segment and so the code inside finally block is executed with/without the presence of try block in java code.

Answer (d)

22. Find output of the following code.

```
public class Test
{
```

```
    private static float temp()
    {
        public static float sum = 21;
        return(--(sum));
    }
    public static void main(String[] args)
    {
        Test test = new Test();
        System.out.println(test.temp());
    }
}
```

a) 21
b) 20
c) Compilation error
d) Runtime error

Ans. (c)

23. State true/false.

We can make a final abstract class in Java. (true/false)

Ans: false

24. State true/false.

We can make a final constructor in Java. (true/false)

Ans: false

25. State true/false.

We can not make a static abstract method in Java. (true/false)

Ans: true