1. Which of the following statements about Java language is true?
   A. Both procedural and OOP are supported in Java.
   B. Java supports only procedural approach towards programming.
   C. Java supports only OOP approach.
   D. None of the above.

2. If an attribute has no access specifier (default), which methods have access to it?
   A. Those defined in the same class.

   B. Static methods in the same class.

   C. Classes in the same package.

   D.   A and C

3. Analyze the code snippet below and determine the output:

```java
package com.core.ct;
class One{
            private One(){
                    System.out.println("One");
            }
            public void display(){
                    System.out.println("Display");
            }
}
public class Two extends One{
            public Two(){
                    System.out.println("Two");
            }
            public static void main(String[] args) {
                    new Two();
                    One obj=new Two();
                    obj.display();

            }
}
```

   A. One   Two   Display

   B. Two   One   Display

   C. One   Display   Two

   D. Compilation Error

4. Analyze the code snippet below and select the missing piece of code to ensure successful compilation:

```
package com.core.ct;
class One{
    public One(String name){

    }
}

public class Two extends One{
        public Two(){
                // Insert Code Here
        }
}
```

A.   One("name")   B. super.One("name")   C.super("name");
     D. this("name");

5. Analyze the code snippet below and determine the output:

```
package com.core.ct;
class One {
    private final void showMessage() {
        System.out.println("One show method");
    }
}
public class Two extends One {
    public final void showMessage() {
        System.out.println("Two show method");
    }
    public static void main(String[] args) {
        Two obj = new Two();
        obj.showMessage();
    }
}
```

   A. One show method   B. Two show method   C. Compilation Error   D. Runtime Error

6. Analyze the code snippet below and determine the output:

```java
package com.core.ct;
class One {
      public void showMessage1() {
            System.out.println("Method-1");
      }
}
public class Two extends One {
      public void showMessage2() {
            System.out.println("Method-2");
      }
      public void showMessage1(){
            System.out.println("Method-3");
      }
      public static void main(String[] args) {
            One ob1 = new Two();
            Two ob2 = (Two)ob1;
            ob2.showMessage1();
            ob2.showMessage2();
      }
}
```

  A. Method - 1
      Method – 2

  B. Method – 2
      Method – 1

  C. Method – 3
      Method – 2

  D. Compilation Error

7. Analyze the code snippet below and determine the output:

```java
package com.core.ct;
class One {
      int a=100;
}
public class Two extends One {
      int a=999;
      public static void main(String[] args) {
            One ob=new Two();
            System.out.println(ob.a);
      }
}
```

  A. 100   B. 999   C. Compilation Error   D. Runtime Exception

8. Analyze the code snippet below and determine the output:

```
package com.core.ct;
final class One {
        public void showMessage1() {
                System.out.println("Method-1");
        }
}
public class Two extends One {
        public void showMessage1(){
                System.out.println("Method-2");
        }
        public static void main(String[] args) {
                One ob1 = new Two();
                ob1.showMessage1();
        }
}
```

    A. Method -1   B. Method – 2   C. Compilation Error   D. Runtime Exception

9. Analyze the code snippet below and determine the output:

```
package com.core.ct;
class One { }
class Two extends One { }

public class TestDemo {
        boolean method1(One one) {
                return true;
        }
        boolean method1(Two two) {
                return false;
        }
        public static void main(String[] args) {
                One ob1 = new One();
                Two ob2 = new Two();
                TestDemo obj = new TestDemo();
                System.out.println(obj.method1(ob1));
                System.out.println(obj.method1(ob2));
        }
}
```

    A. false   true
    B. true   false
    C. The program generates compilation error
    D. The program generates runtime exception

10. Consider the following statements:

1. The keyword 'super' is used by subclass to invoke an overridden method in the superclass
2. Class can have only a constructor
3. For every class , super class is Class
4. Final methods can't be overridden in the subclass

Which of these are true?
A. 2
B. 1 & 2
C. 1 & 4
D. 3

11. Analyze the code snippet below and fill in the blanks to ensure successful compilation.

```java
package com.core.ct;
class One {
    public void showMessage(Object o){
            System.out.println("Method-1");
    }
}
class Two _____ One {
            _____ void showMessage(_____ n){
            System.out.println("Method -2");
    }
}
```
A. implements   protected   String
B. implements   public   String
C. extends   protected   String
D. extends   public   String

12. Analyze the code snippet below and select the missing piece of code to ensure successful compilation:

```
    package com.core.ct;
    class One {
        public int showMessage(int o){
                    return 0;
        }
}
class Two extends One {
        // Insert Code here
}
```

A. public long showMessage(int i){
            return 1;
    }
B. protected int showMessage(int i){
            return 1;
    }
C. private int showMessage(int i){
            return 1;
    }
D. public int showMessage(int i){
            return 1;
    }

13. A class can have many methods with the same name, as long as the number of parameters or type of parameters are different. This OOP concept is known as
        A. Method Invocating
        B. Method Overriding
        C. Method Labeling
        D. Method Overloading

14. Instantiation is completed when
        A. Memory is allocated for a specific object of a class.
        B. java Command is executed
        C. A program is ready for execution.
        D. A program compiles correctly.

15. Inheritance is the process of
        A. Using classes in the established standard Java Language library.
        B. Using features from an existing class.
        C. Combining data and the methods, which process the data, inside the same module.
        D. Splitting a program into multiple related files for each class in the program.

16. What is the output of the following code snippet?
```
 class One{
        One(){
                System.out.print("bike ");
```

```
        }
        One(String s){
                this();
                System.out.print("car ");
        }
}
class Two extends One{
        Two(){
                System.out.print("jeep ");
        }
        Two(String s){
                this();
                System.out.print(s+" ");
        }

}
public class Demo {
        public static void main(String[] args) {
        Two obj=new Two("are under repair");
        }
}
```

A. bike car jeep are under repair
B. jeep bike car are under repair
C. bike jeep car are under repair
D. bike jeep are under repair
E. jeep bike are under repair

17. What is the output of the following code snippet?

```
class Person{
        String name="VP";
        public String speaks(){
                return "speaks Telugu";
        }
}
class Employee extends Person{
        String name="ALN";
        public String speaks(){
                return "speaks Hindi";
        }
}
```

```java
public class Demo {
        public static void main(String[] args) {
                new Demo().getInfo();
        }
        public void getInfo(){
                Person person=new Employee();
                System.out.println(person.name+" "+person.speaks());
        }
}
```

A. ALN speaks Hindi
B. VP speaks Telugu
C. ALN speaks Telugu
D. VP speaks Hindi
E. None of the above

18. What is the output of the following code snippet?
```java
class One{
                static String message="";
                protected One(){
                        message+="I ";
                }
}
class Two extends One{
        private Two(){
                message+="the";
        }
}
public class Demo extends One {
        private Demo(){
                this("am ");
                message+="King ";
        }
        private Demo(String s){
                message+=s;
        }
        public static void main(String[] args) {
                Demo d=new Demo();
                System.out.println(d.message);
        }
}
```

A. I am the King
B. I King
C. I am King
D. Am I king
E. King I am

19. What is the output of the following code snippet?
```java
class One{
```

```java
        public String getMessage(){
                return "I am from one";
        }
}
class Two extends One{
        public String getMessage(){
                return "I am from two";
        }
        public String getGreetings(){
                return "Hi how are you?";
        }
}
public class Demo extends One {
                public static void main(String[] args) {
                                One obj1=new Two();
                                Two obj2=(Two)obj1;
                                System.out.println(obj2.getGreetings());
                                One obj3=new Two();
                                Two obj4=(Two)(One)(Two)obj3;
                                System.out.println(obj4.getMessage());

                }
}
```

   A. Hi how are you?
      I am from One
   B. Hi how are you?
      I am from Two
   C. I am from One
      Hi how are you?
   D. Hi how are you?
   E. None of the above

20. What is the output of the following code snippet?

```
class One{
        int a=100;
        public static String printMessage(){
                return "Hello";
        }
}
class Two extends One{
        int a=200;
        public static String printMessage(){
                return "Hi";
        }
}
public class Demo extends One {
                public static void main(String[] args) {
                                One obj=new Two();
                                System.out.println(obj.a+" "+obj.printMessage());
                }
}
```

    A. 200 Hello
    B. 100 Hi
    C. 100 Hello
    D. 200 Hi
    E. None of the above