1) Overview of Java

```
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, Java!");
    }
}
```

2) Encapsulation, Inheritance, Polymorphism

Encapsulation:

```
class Student {
    private String name;
    public void setName(String name) { this.name = name; }
    public String getName() { return name; }
}
```

Inheritance:

```
class Animal { void sound() { System.out.println("Animal makes sound"); } }
class Dog extends Animal { void sound() { System.out.println("Dog barks"); } }
```

Polymorphism:

```
class Shape { void draw() { System.out.println("Drawing a shape"); } }
class Circle extends Shape { void draw() { System.out.println("Drawing a circle"); } }
```

3) UML Diagram

```
class Animal {
   String name;
   void sound() { System.out.println("Animal makes sound"); }
}

class Dog extends Animal {
   void sound() { System.out.println("Dog barks"); }
}
```

4) Class Creation

```
class Car {
   String model;
}
```

5) Selection Statement (if, switch, if-else)

```
int num = 10;
if (num > 0) { System.out.println("Positive"); }
else if (num < 0) { System.out.println("Negative"); }
else { System.out.println("Zero"); }

switch (num) {
   case 10: System.out.println("Number is 10"); break;
   default: System.out.println("Not 10");  }
```

6) Iteration (while, for, do-while, for-each)

```
int i = 0;
while (i < 3) { System.out.println("While Loop " + i); i++; }

for (int j = 0; j < 3; j++) { System.out.println("For Loop " + j); }

int k = 0;
do { System.out.println("Do-While Loop " + k); k++; } while (k < 3);

int[] arr = {1, 2, 3};
for (int num : arr) { System.out.println("For-each: " + num); }
```

7) Array

```
int[] numbers = {10, 20, 30};
System.out.println(numbers[0]);
```

8) Class, Object, Static Variable, Method, Constructor

```
class Person {
   static int count = 0;  // Static variable
   String name;

   Person(String name) {  // Constructor
      this.name = name;
      count++;
```

```
    }

    void display() {
        System.out.println("Name: " + name);
    }
}
```

9) Method Overloading and Overriding

Overloading:

```
class MathOperations {
    int sum(int a, int b) { return a + b; }
    int sum(int a, int b, int c) { return a + b + c; }
}
```

Overriding:

```
class Parent {
    void show() { System.out.println("Parent class"); }
}
class Child extends Parent {
    void show() { System.out.println("Child class"); }
}
```

10) Constructor

```
class Student {
   String name;
   Student(String name) { this.name = name; }
}
```

11) Variable Length Arguments (Varargs)

```
class VarargsExample {
   void show(int... numbers) {
      for (int num : numbers) {
         System.out.print(num + " ");
      }
   }
}
```

12) Java Inheritance

```
class Parent { void show() { System.out.println("Parent Class"); } }
class Child extends Parent { void show() { System.out.println("Child Class"); } }
```

13) Abstract Class and Abstract Method

```
abstract class Animal {
   abstract void makeSound();
}
class Dog extends Animal {
   void makeSound() { System.out.println("Bark"); }
}
```

14) Interface

```
interface Animal {
   void makeSound();
}
class Dog implements Animal {
   public void makeSound() { System.out.println("Bark"); }
}
```