# Object Oriented Programming with Java (Subject Code: BCS-403)

## Unit 3

## Lecture 23

# Lecture 23

- Static Method
- Base64 Encode and Decode

# Static method in Interface in Java

- Static Methods in Interface are those methods, which are defined in the interface with the keyword static.
- Unlike other methods in Interface, these static methods contain the complete definition of the function and since the definition is complete and the method is static, therefore these methods cannot be overridden or changed in the implementation class.

- Similar to Default Method in Interface, the static method in an interface can be defined in the interface, but cannot be overridden in Implementation Classes.
- To use a static method, Interface name should be instantiated with it, as it is a part of the Interface only.

```
interface NewInterface {
   // static method
    static void hello()
    {
        System.out.println("Hello, New Static Method Here");
    }
   // Public and abstract method of Interface
    void overrideMethod(String str);
}
```

```
// Implementation Class
public class InterfaceDemo implements NewInterface {
    public static void main(String[] args)
    {
        InterfaceDemo interfaceDemo = new InterfaceDemo();
        // Calling the static method of interface
        NewInterface.hello();
        // Calling the abstract method of interface
        interfaceDemo.overrideMethod("Hello, Override Method here");
    }
   // Implementing interface method
     @Override
    public void overrideMethod(String str)
    {
        System.out.println(str);
    }
}
```

# To demonstrate Scope of Static method

- The scope of the static method definition is within the interface only.
- If same name method is implemented in the implementation class then that method becomes a static member of that respective class.

# Program to Demonstrate scope of static method in Interface.

```
interface PrintDemo {
   // Static Method
    static void hello()
    {
        System.out.println("Called from Interface PrintDemo");
    }
}
```

```
public class InterfaceDemo implements PrintDemo {
     public static void main(String[] args)
    {
        // Call Interface method as Interface
        // name is preceding with method
        PrintDemo.hello();
        // Call Class static method
        hello();
    }
    // Class Static method is defined
    static void hello()
    {
        System.out.println("Called from Class");
    }
}
```

# Java Base64 Encode and Decode

- Java provides a class Base64 to deal with encryption.
- We can encrypt and decrypt data by using provided methods.
- We need to import java.util.Base64 in source file to use its methods.
- It uses the Base64 alphabet specified by Java in RFC 4648 and RFC 2045 for encoding and decoding operations.
- The encoder does not add any line separator character. The decoder rejects data that contains characters outside the base64 alphabet.

# Nested Classes of Base64

| Class | Description |
|---|---|
| Base64.Decoder | This class implements a decoder for decoding byte data using the Base64 encoding scheme as specified in RFC 4648 and RFC 2045. |
| Base64.Encoder | This class implements an encoder for encoding byte data using the Base64 encoding scheme as specified in RFC 4648 and RFC 2045. |

# Base64 Methods

| Methods | Description |
|---|---|
| public static Base64.Decoder getDecoder() | It returns a Base64.Decoder that decodes using the Basic type base64 encoding scheme. |
| public static Base64.Encoder getEncoder() | It returns a Base64.Encoder that encodes using the Basic type base64 encoding scheme. |
| public static Base64.Decoder getUrlDecoder() | It returns a Base64.Decoder that decodes using the URL and Filename safe type base64 encoding scheme. |
| public static Base64.Decoder getMimeDecoder() | It returns a Base64.Decoder that decodes using the MIME type base64 decoding scheme. |

# Base64.Decoder Methods

| Methods | Description |
|---|---|
| public byte[] decode(byte[] src) | It decodes all bytes from the input byte array using the Base64 encoding scheme, writing the results into a newly-allocated output byte array. The returned byte array is of the length of the resulting bytes. |
| public byte[] decode(String src) | It decodes a Base64 encoded String into a newly-allocated byte array using the Base64 encoding scheme. |

# Base64.Encoder Methods

| Methods | Description |
|---|---|
| public byte[] encode(byte[] src) | It encodes all bytes from the specified byte array into a newly-allocated byte array using the Base64 encoding scheme. The returned byte array is of the length of the resulting bytes. |
| public int encode(byte[] src, byte[] dst) | It encodes all bytes from the specified byte array using the Base64 encoding scheme, writing the resulting bytes to the given output byte array, starting at offset 0. |

# Basic Encoding and Decoding

```
import java.util.Base64;
public class Base64BasicEncryptionExample {
    public static void main(String[] args) {
        // Getting encoder
        Base64.Encoder encoder = Base64.getEncoder();
        // Creating byte array
        byte byteArr[] = {1,2};
        // encoding byte array
        byte byteArr2[] = encoder.encode(byteArr);
        System.out.println("Encoded byte array: "+byteArr2);
        byte byteArr3[] = new byte[5];
        int x = encoder.encode(byteArr,byteArr3);
        System.out.println("Encoded byte array written to another array: "+byteArr3);
        System.out.println("Number of bytes written: "+x);
      }}
```

# Output

```
Encoded byte array: [B@15db9742
Encoded byte array written to another array: [B@6d06d69c
Number of bytes written: 4
```

# Basic Encoding and Decoding

```
import java.util.Base64;
public class Base64BasicEncryptionExample1 {
    public static void main(String[] args) {
         // Getting encoder
        Base64.Encoder encoder = Base64.getEncoder();
        // Encoding string
        String str = encoder.encodeToString("ABES Engineering
College".getBytes());
        System.out.println("Encoded string: "+str);
        // Getting decoder
        Base64.Decoder decoder = Base64.getDecoder();
        // Decoding string
        String dStr = new String(decoder.decode(str));
        System.out.println("Decoded string: "+dStr);
    }}
```

# Output

```
Encoded string: QUJFUyBFbmdpbmVlcmluZyBDb2xsZWdl
Decoded string: ABES Engineering College
```

# URL Encoding and Decoding

```
import java.util.Base64;
public class Base64URLEncryptionExample {
    public static void main(String[] args) {
        // Getting encoder
        Base64.Encoder encoder = Base64.getUrlEncoder();
        // Encoding URL
        String eStr = 
encoder.encodeToString("https://abes.ac.in/".getBytes());
        System.out.println("Encoded URL: "+eStr);
        // Getting decoder
        Base64.Decoder decoder = Base64.getUrlDecoder();
        // Decoding URl
        String dStr = new String(decoder.decode(eStr));
        System.out.println("Decoded URL: "+dStr);
    }
}
```