

Midterm Homework 2

I.

1. interface
2. `octalString = Integer.toOctalString(decimalNumber);`
3. `binaryString = Integer.toBinaryString(decimalNumber);`
4. `hexadecimalString = Integer.toHexString(decimalNumber);`
5. package numbers;

II. On your document explain how the programs may be modified/improved so that decimal numbers with the decimal point may also be converted. If necessary, include/show codes/program segment in your explanation.

1. Change the data type of decimalNumber to double.

```
private double decimalNumber; //decimal number form
```

2. Create new methods that will return a binary, octal or hexadecimal string version of a decimal number with a decimal point;
`toBinaryString()`, `toOctalString()`, `toHexadecimalString()`.

```
public static String toBinaryString(double decimal) {  
    String binary, fractionalBinary = ".";  
    double fractional = decimal % 1, dividend;  
  
    binary = "" + Integer.toBinaryString((int) (decimal));  
  
    do {  
        dividend = fractional * 2;  
        fractional = dividend % 1;  
        fractionalBinary = fractionalBinary + (int) dividend;  
    } while (dividend % 1 != 0);  
  
    if (decimal % 1 != 0) {  
        binary = binary + fractionalBinary;  
    }  
    return binary;  
}
```

Convert the whole part to Binary String using the Integer class.

Convert also the fraction part into its binary equivalent.

If the decimal has a fraction part then concatenate it to the binary string of the whole part.

Same algorithm is applied to the other methods.

```

    }

    public static String toOctalString(double decimal){
        String octal, fractionalOctal = ".";
        double fractional = decimal%1, dividend;
        octal = "" + Integer.toOctalString((int)(decimal));
        do {
            dividend = fractional * 8;
            fractional = dividend%1;
            fractionalOctal = fractionalOctal + (int)dividend;
        } while(dividend%1 != 0);
        if(decimal%1 != 0){
            octal = octal + fractionalOctal;
        }
        return octal;
    }

    public static String toHexadecimalString(double decimal){
        String hexa, fractionalHex = ".";
        double fractional = decimal%1, dividend;
        hexa = "" + Integer.toHexString((int)(decimal));
        do {
            dividend = fractional * 16;
            fractional = dividend%1;
            fractionalHex = fractionalHex + hexDigit((byte) dividend);
        } while(dividend%1 != 0);
        if(decimal%1 != 0){
            hexa = hexa + fractionalHex;
        }
        return hexa;
    }
}

```

3. Replace the static methods of integer class (that gets the binary, octal and hexadecimal string of a decimal) with invocations of the newly created methods in number 2.

```

public void setDecimalNumber(double d) {
    decimalNumber = d;
    hexadecimalString = toHexadecimalString(decimalNumber);
    binaryString = toBinaryString(decimalNumber);
    octalString = toOctalString(decimalNumber);
}

```

```

public void setHexadecimalString(String n) {
    hexadecimalString = n;
    decimalNumber = hexadecimalToDecimal(hexadecimalString);
    binaryString = toBinaryString(decimalNumber);
    octalString = toOctalString(decimalNumber);
}

```

```

public void setOctalString(String n) {
    octalString = n;
    decimalNumber = octalToDecimal(octalString);
    binaryString = toBinaryString(decimalNumber);
    hexadecimalString = toHexadecimalString(decimalNumber);
}

```

```

public void setBinaryString(String b) throws Exception {
    binaryString = b;
    decimalNumber = binaryToDecimal(binaryString);
    octalString = toOctalString(decimalNumber);
    hexadecimalString = toHexadecimalString(decimalNumber);
}

```

4. Create a new Boolean method hasDecimalPoint() that returns true if a decimal, octal, or hexadecimal number has a decimal point.

```

public boolean hasDecimalPoint(String a){
    boolean hasDecimal = false;
    for(int i = 0; i<a.length(); i++){
        if(a.charAt(i)=='.'){
            hasDecimal = true;
        }
    }
    return hasDecimal;
}

```

5. Add '.' in the isValidBinaryString() so it doesn't catch an error.

```

public boolean isValidBinaryString(String s) {
    boolean result = true;
    for (int index = 0; index < s.length() && result; index++) {
        if (s.charAt(index) != '0' && s.charAt(index) != '1' && s.charAt(index) != '.')
            result = false;
    }
    return result;
}

```

6. Replace int with double.

```

public interface NumberConverter{
    public double binaryToDecimal(String b) throws Exception;
    public double hexadecimalToDecimal(String h);
    public double octalToDecimal(String n);
}

```

7. Replace the return type to double and modify binaryToDecimal(), octalToDecimal(), and hexadecimalToDecimal() to include the decimal parts. Use the method hasDecimalPoint(). If the binary, octal or hexadecimal number has a decimal point then get the decimal part by splitting the string with ".". The decimal part will then be converted to its corresponding string value and added to the previously converted result.

An if statement was added to the method binaryToDecimal() method.

```
public double binaryToDecimal(String b) throws Exception {
    double result = 0;
    if (!isValidBinaryString(b))
        throw new Exception("Invalid Binary Number");

    if (hasDecimalPoint(b)) {
        String data[] = b.split( s: "." );
        for (int i = 0; i < data[0].length(); i++) {
            result += Integer.parseInt( s: "" + data[0].charAt(i) * Math.pow(2, (int) (data[0].length() - 1 - i)));
        }
        for (int i = 0; i < data[1].length(); i++) {
            result += Integer.parseInt( s: "" + data[1].charAt(i) * Math.pow(2, (int) (-data[1].length() - i)));
        }
    } else {
        for (int i = 0; i < b.length(); i++) {
            result += Integer.parseInt( s: "" + b.charAt(i) * Math.pow(2, (int) (b.length() - 1 - i)));
        }
    }
    return result;
}
```

The following if statement was added to the method hexadecimalToDecimal().

```
if (hasDecimalPoint(h)) {
    String data[] = h.split( s: "." );
    for (int i = 0; i < data[1].length(); i++) {
        switch (data[1].charAt(i)) {
            case '0':
            case '1':
            case '2':
            case '3':
            case '4':
            case '5':
            case '6':
            case '7':
            case '8':
            case '9':
                result += Integer.parseInt( s: "" + data[1].charAt(i) * Math.pow(16, (int) (-data[1].length() - i)));
                break;
            case 'a':
            case 'A':
                result += 10 * Math.pow(16, (int) (-data[1].length() - i));
                break;
            case 'b':
            case 'B':
                result += 11 * Math.pow(16, (int) (-data[1].length() - i));
                break;
            case 'c':
            case 'C':
                result += 12 * Math.pow(16, (int) (-data[1].length() - i));
                break;
            case 'd':
            case 'D':
                result += 13 * Math.pow(16, (int) (-data[1].length() - i));
                break;
            case 'e':
            case 'E':
                result += 14 * Math.pow(16, (int) (-data[1].length() - i));
                break;
            case 'f':
            case 'F':
                result += 15 * Math.pow(16, (int) (-data[1].length() - i));
                break;
        }
    }
}
return (result);
}
```

An if statement was added to the octalToDecimal() method.

```
public double octalToDecimal(String n) {
    double result = 0;
    for (int i = 0; i < n.length(); i++) {
        result += Integer.parseInt( "" + n.charAt(i)) * Math.pow(8, (int) (n.length() - 1 - i));
    }
    if(hasDecimalPoint(n)){
        String data[] = n.split( "\\." );
        for (int i = 0; i < data[1].length(); i++) {
            result += Integer.parseInt( "" + data[1].charAt(i)) * Math.pow(8, (int) (-data[1].length() - i));
        }
    }
    return (result);
}
```

Here is a sample output run of the modified code that allows decimal with decimal point to be converted.

Number Converter

This application helps you convert a decimal number to binary, octal, and hexadecimal. Please enter the decimal number in the field provided. Then, press the convert key.

Decimal Number:	59.625
Binary Number:	111011.101
Octal Number:	73.5
Hexadecimal Number:	3b.A

convert clear exit