

CheckWriter.java

```
////////////////////////////////////
//www.karamian.com
//Program CheckWriter
////////////////////////////////////

import java.io.*;
import java.util.*;
import java.text.Format.*;

//*****
// Class CheckWriter
//*****

public class CheckWriter
{
    String digits[]={ "", "ONE ", "TWO ", "THREE ", "FOUR ", "FIVE ", "SIX ", "SEVEN
", "EIGHT ", "NINE " };
    String tens[]={ "", "TEN ", "TWENTY ", "THIRTY ", "FORTY ", "FIFTY ", "SIXTY ",
"SEVENTY ",
                "EIGHTY ", "NINETY " };
    String teens[]={ "TEN ", "TEN ", "ELEVEN ", "TWELVE ", "THIRTEEN ", "FOURTEEN ",
"FIFTEEN ",
                "SIXTEEN ", "SEVENTEEN ", "EIGHTEEN ", "NINETEEN
"};
    String placeHolder[]={ "MILLION ", "THOUSAND ", "", "", "" };
    String moneyWord;
    java.text.DecimalFormat twoDigit;
    java.text.DecimalFormat prepInt;

    //=====
    // The CheckWriter constructor.
    //=====

    public CheckWriter()
    {
        twoDigit=new java.text.DecimalFormat("0.00");
        prepInt=new java.text.DecimalFormat("00");
        moneyWord="";
    } // end CheckWriter constructor

    //=====
    // The createWord method.
    // Recursively generates word value by splitting the number into sizes of one
hundred, also
    // checks to ensure that a value such as 1000000 does not generate a wrong word
value
    // return.
    //=====

    public String createWord(int money, int divisor, int n)
    {
        if(divisor==0)
        {
            return moneyWord;
        }
        else if(money<1000)
        {
            int tempMoney=money/divisor;
            moneyWord=moneyWord.concat(oneHundred(tempMoney));
            return createWord(money%divisor, divisor/1000, n+1);
        }
        else
        {
            int tempMoney=money/divisor;
```

```

                                CheckWriter.java
    moneyWord=moneyWord.concat(oneHundred(tempMoney).concat(placeholder[n]));
    return createWord(money%divisor, divisor/1000, n+1);
}

} // end createWord method

//=====
// The oneHundred method.
// Takes value from createWord method and generates appropriate word value for
numbers
// under 100 and returns this value back to creatWord method.
//=====

public String oneHundred(int tempDollar)
{
    String tempMoneyWord="";
    if(tempDollar>=100)
    {
        tempMoneyWord=digits[tempDollar/100].concat("HUNDRED ");
        tempDollar=(tempDollar%100);
    }
    if(tempDollar>=20 || tempDollar<10)
    {
        String s=tens[tempDollar/10];
        String t=digits[tempDollar%10];
        tempMoneyWord=tempMoneyWord.concat(s.concat(t));
        return tempMoneyWord;
    }
    else
    {
        String u=teens[tempDollar%10];
        tempMoneyWord=tempMoneyWord.concat(u);
        return tempMoneyWord;
    }
}

} // end oneHundred method.

//=====
// The printCheckWriter method.
// Converts double to int and seperates cents from dollars and sets divisor for
createWord
// method and calls createWord method to generate word output of money amount.
//=====

public String printCheckWriter(double dedollar)
{
    moneyWord="";
    double cent=((dedollar*100)%100);
    double tmpDollar=(dedollar-(cent/100));
    String iCent=prepInt.format(tmpDollar);
    String g=prepInt.format(cent);
    int dollar=Integer.parseInt(iCent);
    if(dollar>=1000000)
    {
        moneyWord=createWord(dollar, 1000000, 0).concat("DOLLARS AND
").concat(g.concat(" CENTS"));
        return moneyWord;
    }
    else if(dollar<1000000&& dollar>=1000)
    {
        moneyWord=createWord(dollar, 1000, 1).concat("DOLLARS AND
").concat(g.concat(" CENTS"));
        return moneyWord;
    }
}

```

CheckWriter.java

```
    else
    {
        moneyWord=oneHundred(dollar).concat("DOLLARS AND ").concat(g.concat("
CENTS"));
        return moneyWord;
    }

} // end printCheckWriter method

//=====
// The fileCheckWriter method.
// Imports a file book and outputs to file customer numeric and word format of
numbers
// listed in file.
//=====

public void fileCheckWriter() throws IOException
{
    String s;
    BufferedReader inPS=new BufferedReader(new InputStreamReader(
new FileInputStream(
        new File("book"))));
    PrintStream outPS=new PrintStream(new FileOutputStream(new File("customer")));
    while((s=inPS.readLine())!=null)
    {
        double g=Double.parseDouble(s);
        outPS.println(twoDigit.format(g)+ " "+ printCheckWriter(g));
    }
} // end fileCheckWriter method
} //end of Class CheckWriter.
```