

2.2 Method 2: Verification Program for Java/C++/Others

Since you cannot compile the Java source code to machine code and you can compile Java source to byte-codes you cannot use the technique given in Method 1 above. When you do diff on Java class files it will always be different.

In this method, a different technique will be given which can be used to validate any beautifier program for Java. Also this method is quite powerful and can be used to validate any beautifier program for any language like C, C++, PERL, SQL, HTML or Java. Since all beautifier program simply rearrange or insert whitespaces, you can strip all the whitespaces from original source file and dump it to a file called verify1.out and strip all the whitespaces from beautified source file and dump it to a file called verify2.out. Now, do a diff on verify1.out and verify2.out. If there is no difference, then beautifier program is working properly. The method is not 100% perfect and can catch atleast 98% of the errors/bugs in the beautifier program. Use this method in conjunction with other methods. But this method is better than not having a verification at all and blindly trusting the beautifier program!!

Note: A whitespace can be one of following – blank space ' ', form-feed '\f', newline '\n', carriage return '\r', horizontal tab '\t' or vertical tab '\v'.

```
bash$ java StripWhitespaces sample.java > verify1.out
bash$ java StripWhitespaces sample_beutified.java > verify2.out
bash$ diff verify1.out verify2.out

bash$ java StripWhitespaces sample.cpp > verify1.out
bash$ java StripWhitespaces sample_beutified.cpp > verify2.out
bash$ diff verify1.out verify2.out

bash$ java StripWhitespaces sample.sql > verify1.out
bash$ java StripWhitespaces sample_beutified.sql > verify2.out
bash$ diff verify1.out verify2.out
```

The source code of StripWhitespaces Java program is not given here. It is left as an exercise for students (you) to write a small program in Java which will simply strip whitespaces from the input text file and output to standard console output. Students are also urged to write this small program (StripWhitespaces) in C, PERL, Unix shell script (Korn, Bourne) and AWK script. Students can see howto the same task can be accomplished in these five different languages and can do comparison of ease of programing. You should *put a newline '\n' character after every 50 characters* while generating verify1.out and verify2.out so that when you do a diff you can see on which lines differences are coming up. Otherwise, verify*.out files will just contain one line and it will be difficult to pin-point where exactly the beautifier program is failing (got this point ???).

2.3 Method 3: Shell script: Verification Program for C++/C

This is a Korn shell script to verify beautifier program. Requires "pdksh*.rpm" from Linux 'contrib' cdrom. Save this file as 'text' file and chmod a+rx on it. You can re-write this shell script in PERL so that you can use it on Window 95/NT or MSDOS. Uncomment the PRGM variable to point to **bcpp**, **cb** or **indent**

```
#!/bin/ksh
```

C-C++ Beautifier HOW-TO

```
# Verification program to check C++ Beautifiers 'bcpp', 'indent' or cb
#####
# Copyright
# The copyright policy is GNU/GPL.
# Author: Al Dev (Alavoor Vasudevan) alavoor[AT]yahoo.com
#####

check_beautify_now()
{
    # Remove all the temp files....
    \rm -f ${TMP_FILE}
    \rm -f ${TMP_CPPFILE}.*

    FNAME=$1
    if [ ! -f ${FNAME} ]; then
        print "\nError: The file ${FNAME} does not exist!!. Aborting now ...."
        exit
    fi
    \cp -f ${FNAME} ${TMP_CPPFILE}.cpp
    ${COMPILER} -c ${TMP_CPPFILE}.cpp
    if [ ! -f ${TMP_CPPFILE}.o ]; then
        print "Fatal Error: Failed to compile ${FNAME}. Aborting now... "
        exit
    fi
    \mv -f ${TMP_CPPFILE}.o ${TMP_CPPFILE}_orig.o

    aa=`basename $PRGM`
    print "\nRunning, verifying $aa on ${FNAME}"
    ${PRGM} ${TMP_CPPFILE}.cpp
    ${COMPILER} -c ${TMP_CPPFILE}.cpp
    \rm -f $TMP_FILE
    diff ${TMP_CPPFILE}.o ${TMP_CPPFILE}_orig.o 1> $TMP_FILE 2>> $TMP_FILE
    result=""
    result=`wc -c $TMP_FILE | awk '{print $1}'`
    if [ "$result" = "0" ]; then
        print "Success!! Beautifier $aa is working properly!!\n"
    else
        print "Fatal Error: Something wrong!! Beautifier is not working!!"
        exit
    fi
    #
    ${COMPILER} -S ${TMP_CPPFILE}.cpp
    #
    diff ${TMP_CPPFILE}.s ${TMP_CPPFILE}_orig.s

    # Remove all the temp files....
    \rm -f ${TMP_FILE}
    \rm -f ${TMP_CPPFILE}.*
}

##### Main of program begins here #####3
PRGM=/usr/bin/bcpp
PRGM=/usr/bin/cb
PRGM=/usr/bin/indent
COMPILER=/usr/bin/g++

TMP_FILE=beautify.tmp
TMP_CPPFILE=beautify-tmp_cppfile

print -n "Enter the C++ file name <default is *.cpp> : "
read ans
if [ "$ans" = "" -o "$ans" = " " ]; then
    ans="ALL"
else
```