MICHAEL LOWE
COP 4600
LAB 4 DEBUGGER

Context: Given the program, divint.cc my assignment was to find the bug, which caused my program to crash and draw the following error (below).

```
[bash-3.2$ ls                                   ]
 div                run
 div.dSYM           run.dSYM
 divint.cc
[bash-3.2$ ./div                                ]
 Floating point exception: 8
 bash-3.2$
```

After installing GNU's GDB program and compiling my program with the –g flag in the following manner:

G++ -g divint.cc –o div

Exploring further I was able to discover using my GDB tools break at the print after each function call, I realized there was an issue with the second function call. To better see the outputs, I modified the code to print on each line(just for my own sanity)

```
int divint(int, int);
int main() {
   int x = 5, y = 2;
   cout << divint(x, y) << endl;

   x = 3; y = 0;
   cout << divint(x, y) << endl;

   return 0;
}

int divint(int a, int b) {
   return a / b;
}
```

After setting my breakpoints at the end of each function call, in GDB I ran into an error because I was using my mac to do this project. I had to trouble shoot it to get

the correct output from GDB—using commands break 10 and break 13, then running the program with run div

Once running, I realized that the error was triggered at line 12 with the function calling a request to divide by zero.

```
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from div...Reading symbols from /Users/michaellowe/Desktop/TEMP/EX4/div.
dSYM/Contents/Resources/DWARF/div...done.
done.
(gdb) break 13
Breakpoint 1 at 0x100000d0f: file divint.cc, line 13.
(gdb) run
Starting program: /Users/michaellowe/Desktop/TEMP/EX4/div
Unable to find Mach task port for process-id 2270: (os/kern) failure (0x5).
 (please check gdb is codesigned - see taskgated(8))
```

Once handled, the program ceased to crash, and the bug was handled.

```
#include <iostream>
using namespace std;

// Shout out to www.tutorialspoint.com

int divint(int, int);
int main() {
   int x = 5, y = 2;
   cout << divint(x, y) << endl;
   // (gdb) break 10
   x = 3; y = 1; // not a division by ZERO
   cout << divint(x, y) << endl;
   //(gdb) break 13
   return 0;
}

int divint(int a, int b) {
   return a / b;
}
```

```
bash-3.2$ g++ -g divint.cc -o div
bash-3.2$ ./div
2
3
bash-3.2$
```