PART 1 - From Professor Kent Chin - div_casting.cpp

/* Literals are FIXED values (e.g. 0, 5, -2, 3.14)
   Whole-number literals (e.g. 0, 1, -3) are integer types
   Literals with decimal points (e.g. 3.14, 2.718) are float/double types */

What’s the output for a - e below?

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

int main() {
    int x = 5; double y = 3;

    // Integer division rounds the result down to the nearest whole number.
    cout << "1a: " << x / 3 << endl; //a

    // Casting converts a variable or literal from one data type to another
    cout << "1b: " << (int)9.73 << endl; //b
    cout << "1c: " << (double)22 / 7 << endl; //c

    // What happens when division between two integers happened BEFORE casting?
    cout << "1d: " << (double)(22 / 7) << endl;  //d

    // Any arithmetic operation with a double value will result in a double // value.
    cout << "1e: " << y / x << endl; //e
    return 0;
}
```

PART 2 – Consider the following C++ program.

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

// line a)
int i = 7.5; double d = 2;
    cout << i / d << endl; // line a)
    cout << (double) i / d << endl; // line b)
    cout << i / (int) d << endl; // line c)
    cout << 7 / 2 << endl; // line d)
    cout << 15.0 / 4.0 << ' ' << 2.2 + 5 << endl; // line e)
    cout << 5 * 2.0 << endl; // line f)
    cout << 5 % 3 << endl; // line g)
    cout << 1 + 4 % 5 * 3 << endl; // line h)
return 0;
```

State the output at line:

<table>
<thead>
<tr>
<th>a)</th>
<th>b)</th>
<th>c)</th>
<th>d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>e)</td>
<td>f)</td>
<td>g)</td>
<td>h)</td>
</tr>
</tbody>
</table>
CS 111 Lab – Arithmetic, variable types, if-statement

Part 3 - Which variable type stores each of the following:

a. single character __________________
b. whole number __________________
c. decimal number __________________
d. true or false value __________________
e. text longer than one character _________________

Part 4 - Change in bills.
Write a program that prompts the user for the amount of change in bills that he/she needs. If the user provides a negative amount, print “Invalid amount!” to the screen and exit the program. Otherwise, the program is to print the amount of change (in bills) with hundreds, twenties, tens, fives, and ones (yes, there are other bills out there, but we'll only worry about these bills).

Note: You can use return 0 or exit(1) for program termination. If you are using exit(1), remember to add this to the beginning of the program: #include <cstdlib>

HINT: Feel free to use the “Change In Coins” program from lecture as a reference. And remember to use variable names that make sense.

Sample I/O
How much change in dollars is needed? 999
You need:
9 hundred dollar bill(s)
4 twenty dollar bill(s)
1 ten dollar bill(s)
1 five dollar bill(s)
4 one dollar bill(s)

How much change in dollars is needed? 567
You need:
5 hundred dollar bill(s)
3 twenty dollar bill(s)
0 ten dollar bill(s)
1 five dollar bill(s)
2 one dollar bill(s)

How much change in dollars is needed? -231
Invalid amount!