CS 007A, Fall 2012 – Midterm 1, Chapters 1—3 Solutions.

1. Which of the following declarations are illegal, and why? Be sure to state your reasons.
   a. `int float;`
      ANS: This is illegal since `float` is a keyword and so can't be a variable name.
   b. `float 33.7;`
      ANS: This is illegal since a variable name can't start with a number.
   c. `bool end?;`
      ANS: This is illegal since the question mark can't be part of a variable name.
   d. `unsigned next-number;`
      ANS: This is illegal since the dash can't be part of a variable name.

2. Each of the following programs has an error. Locate the error, classify the error as either a syntax error, or a logical error and then correct it.:
   a.
```
#include <iostream>;
using namespace std;
int main() {
   int n = 17;
   cout << n/2 << endl; }
```
   ANS: The semicolon at the end of "`#include <iostream>;`" is a syntax error.
   b.
```
#include <iostream>
using namespace std;
int main() {
   float n /= 10.1;
   cout << n*10.1: }
```
   ANS: The problem is with "`float n /= 10.1;`" Since `n` hasn't been initialized, dividing it by `10.1` is a logic error. There is also a colon where there should be semicolon (last line) – how did that get there?

3. Write a loop that will cause an integer overflow error in any compiler.
   ANS:
```
#include <iostream>;
using namespace std;
int main() {
   int n = 10;
   while(1) n *= 10;
   return 0; }
```

4. Determine the value of the following expressions, assuming `a=5`, `b=2`, `c=4` and `d=5`
   a. `a==5`
      ANS: "true," or "1."
   b. `b + d == c * c`
      ANS: 7 == 16 is "false," or "0."
   c. `d % b * c > 5 || c % b * d < 7`
      This is equivalent to
```
((5 % 2) * 4 > 5) || ((4 % 2) * 5 < 7)
(1 * 4 > 5) || (0 * 5 < 7)
(4 > 5) || (0 < 7)
0 || 1 = 1, or "true"
```

5. What is the output of the following program if the user inputs 1600?

```
#include <iostream>
using namespace std;
int main() {
   int murbs, hurbs, durbs, x;
   cout << "\nInput a an integer: ";
   cin >> x;
   durbs = x/(24*60);
   hurbs = x%(24*60)/60;
   murbs = x%(24*60)%60;
   cout << endl << x << " murbs is equivalent to " << durbs
        << " durbs " << hurbs << " hurbs and "
        << murbs << " murbs.\n";
   return 0; }
```

ANS:

`durbs = 1600/1440` is equivalent to `10/9` whose integer value is `1`.
`hurbs = 1600%(1440)/60` is evaluated left to right, so we get a remainder of `160`, which is then divided by 60 to give the value `2`.
`murbs = 1600%(1440)%60` is also evaluated left to right to we have `160%60` is `40`.
Thus the output is `"1600 murbs is equivalent to 1 durbs 2 hurbs and 40 murbs."`

Note that this makes more sense if you substitute the words "minutes" for "murbs," "hours" for "hurbs," and "days" for "durbs."

6. Consider the following program fragment.

```
int accumulator = 0, sam, pam;
cout << "\nEnter integers for sam and pam: ";
cin >> sam >> pam;
while (true ) {
   if (pam == 0) break ;
   accumulator += ((pam % 2 == 1) ? sam : 0);
   pam /= 2;
   sam *= 2;  }
cout << accumulator << "\n";
```

a. Complete the following tables until the program completes, or indicate that it's an infinite loop:

sam	pam	accumulator
5	4	0
10	2	0
20	1	0
40	0	20

sam	pam	accumulator
6	17	0
12	8	6
24	4	6
48	2	6
96	1	6
192	0	102

b. In a few words, describe the output of this program in terms of the input values of `sam` and `pam`, and how it works.

ANS: This is a multiplication algorithm that uses the binary form of the second factor like so:
$5 * 4 = 5 * (2^2) = 20$ so just double 5 twice. This is recognized by dividing by writing $2^2 = 100_2$ and proceding thus: $5 * 100_2 = 10 * 10_2 = 20 * 1 = 20$. Similarly,

$$6 * 17 = 6 * 10001_2 = 6 + 12 * 1000_2 = 6 + 24 * 100_2 = 6 + 48 * 10_2 = 6 + 96 * 1 = 102$$

or $3 * 21 = 3 * 10101_2 = 3 + 6 * 1010_2 = 3 + 12 * 101_2 = 15 + 24 * 10_2 = 15 + 48 = 63$.

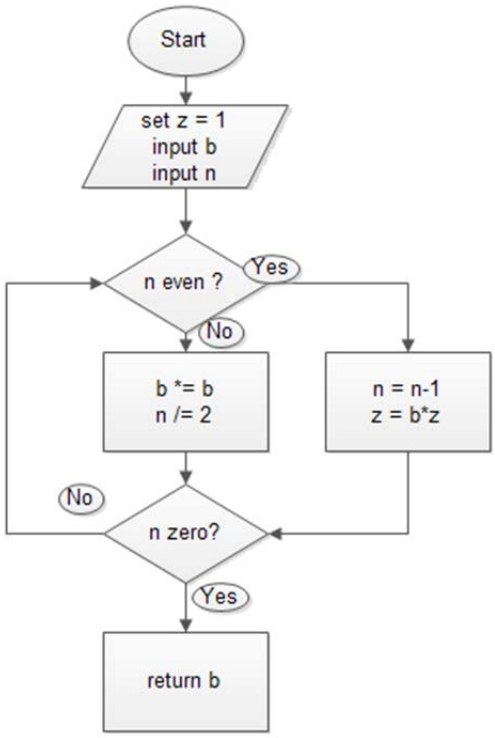
7. Write a C++ program to implement the flow chart below.

```
// implementation of flow chart

#include <iostream>

using namespace std;

int main()
{
   int b,n,z = 1;
   cout << "\nInput b and n and we'll compute
        << something with them: ";
   cin >> b >> n;
   cout << "\nb = " << b << "\nn = " << n;
   while(n!=0) {
          if(n%2==0) // n is even
          {
                 --n;
                 z *= b;
          }
          else
          {
                 b *= b;
                 n /= 2;
          }
   }
   cout << "\nThe result is " << b << endl;
}
```



8. Consider the following program?

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

int main()
{
   int i = 5;
   for(int i = 1; i < 10; ++i)
   {
         cout << i*i << '\t';
         if(i%3 == 0) cout << endl;
   }
   cout << "\ni = " << i << endl;
   return 0;
}
```

a. What is the output of the program?

ANS:

```
1       4       9
16      25      36
49      64      81

i = 5
```

b. Rewrite the for loop as an equivalent while loop and so that the output of the program doesn't change.
ANS: The trick is to just leave the i with outer scope alone and use another variable, say, j, for the while loop.

```
#include <iostream>
using namespace std;
int main() {
      int i = 5, j = 1;
      while(j < 10)     {
            cout << j*j << '\t';
            if(j%3 == 0) cout << endl;
            ++j;  }
      cout << "\ni = " << i << endl;
      return 0; }
```

9. The sum of the first *n* cubes is given by $1 + 2^3 + 3^3 + \cdots + n^3 = \left(\frac{n(n+1)}{2}\right)^2$. Write a complete program that checks this formula by inputting *n* and then computing and comparing the values of both sides of the equation. Do not use the `cmath` library.
ANS:

```
#include <iostream>
using namespace std;
int main() {
   int n, sum = 0;
   cout << "\nTo add the first n cubes, 1 + 8 + 27 + ...."
        << "\nEnter n: ";
   cin >> n;
   for(int i = n; i >= 0; --i)  {
        sum += i*i*i;     }
   cout << "\nThe sum of the first " << n << " cubes is " << sum << endl;
   cout << "\n(n*(n+1)/2)^2 = " << n*n*(n+1)*(n+1)/4 << endl;
   return 0; }
```

10. Write a program that adds the computes the smallest value of *n* so that $1 + \frac{1}{8} + \frac{1}{27} + \cdots + \frac{1}{n^3} > 1.2$
Do not use the `cmath` library.

```
#include <iostream>
using namespace std;
int main() {
      double n = 1, sum = 0;
      while(sum <= 1.2) {
            sum += 1/(n*n*n);
            ++n;  }
      cout << "\nAfter " << n
           << " terms, 1 + 1/8 + 1/27 + ... + 1/" << n
           << "^3 = " << sum << " > 1.2" <<  endl;
      return 0; }
```

After 17 terms, 1 + 1/8 + 1/27 + … + 1/17^3 = 1.20022 > 1.2