CSCE 2004 – Midterm Exam Practice Problem – Fall 2016

The practice problems are selected from actual midterm questions in previous midterm exams. Only those questions that can be solved using the material studied or covered are selected.

It is by no means that these questions or variations of these questions will be used as questions for our midterm.

These questions illustrate the expectation we have for our students. These questions are used to assess whether students have acquired the needed knowledge, developed the fundamental understanding of computing, and gained skills in computer programming in C++ in the Linux environment.

I. Data types

1.) Define the following data types, and give an example of each one.

Type	*Definition (what it is for)*	*Example*
int		
double		
char		
bool		
string		

2a.) What is the output of the following code?

```
int x = 11;
int y = 10;
float z = x/y;
cout << z;
```

2b.) What is the output of the following code?

```
int x = 11;
int y = 10;
float z = x/(float)y;
cout << z;
```

2c.) Why are the answers for 2a and 2b different? What is the name of the technique used to make them different?

II. Boolean expressions

Consider the following Boolean expression: (A && B) || (B && C).

We can determine exactly what values of A, B, and C make this expression true or false by building a truth table. The column titled !C is not necessary here, but it shows you how to negate a variable. Simply change all the true values to false, and all the false values to true.

A	B	C	!C	A && B	B && C	(A && B) || (B && C)
T	T	T	F	T	T	T
T	T	F	T	T	F	T
T	F	T	F	F	F	F
T	F	F	T	F	F	F
F	T	T	F	F	T	T
F	T	F	T	F	F	F
F	F	T	F	F	F	F
F	F	F	T	F	F	F

So for example, when A = false, B = true, and C = true, the last expression on the right is also true.

Complete the following truth table.

A	B	C	A && B	B || C	(A && B) || (B || C)	!((A && B) || (B || C))
T	T	T				
T	T	F				
T	F	T				
T	F	F				
F	T	T				
F	T	F				
F	F	T				
F	F	F				

III. Loops

This for loop does not do what the comment says it should do.

```
//This loop should print the numbers 1 through 10.
for(int x = 1; x >= 10; x--)
     cout << x << endl;
```

1a.) What happens when we try to run this loop?

1b.) What changes should be made to make it run correctly (hint: we need 2 changes)?

2.) What is the primary difference between a for loop and a while loop, in terms of how many times each loop runs?

IV. Arrays

1.) We have a 1x10 integer array called numbers. Write some code that will scan through the array and print all the EVEN numbers.

For example, if your array looks like this:

12	13	54	356	36	67	2	34	98	0

Your output should be:

12 54 356 36 2 34 98 0

2.) We have a 1x10 integer array called numbers. Write some code that will reverse the array **in place**. That is, do not create a second array and simply fill it in reverse order.

For example: if the array originally looks like this:

1	2	3	4	5	6	7	8	9	10

The final product should look like this:

10	9	8	7	6	5	4	3	2	1

If-Statements (12 points)

Given the following code, answer the following three questions.

```
string choice;
double money;
cout << "Enter the amount of money you have: ";
cin >> money;
cout << "Enter 'd' to divide by 3, 'm' to multiply by 2.7 or
'a' to add 0.36: ";
cin >> choice;

//  THE CODE FOR PARTS b and c GOES HERE

if (choice == "d")
   money = money / 3.0;
else if(choice == "m")
   money = money * 2.7;
else if (choice == "a")
   money = money + 0.36;
```

1. Assume the user enters 36 and then 'd'. What will be the value of "money" be *after* the if statements?

a) 97.2
b) 36.36
c) 12.0
d) 36.0
e) None of the above

2. Write an if statement that prints out "Valid" if the money entered is not less than 50.0 or greater than 75.0, inclusive of 50.0 and 75.0.

3. Write an if statement that prints out "Valid" if the user enters 'd', 'm', or 'a'.

Loops (16 points)

1. How many times will the following loop display "Count!"?

```
for (int i = 1; i < 67; i=i+1)
{
   cout << "Count!" << endl;
}
```

a) 67
b) 68
c) 66
d) 0
e) None of the above

2. What is the output of the following code segment?

```
int n = 2;
while (n > 1 &&  n <= 5)
{
   cout << n << " ";
   n = n+3;
}
```

a) 2 5
b) 2 3
c) 2 3 4 5
d) 2 2 2 ... and so on forever
e) None of the above

3. What will the following code display?

```
int x = 0;
for (int count = 3; count >= 0; count=count-1)
    cout << x << " ";
```

4. Consider the following C++ program fragment:

```
int i, j, count = 2;
i = 3;
while (i > 1)
{
      for (j = 0; j <= 4; j = j + 2)
      {
            count=count+1;
      }

      i=i-1;
}
```

What is the final value of "count"?

Miscellaneous True/False (6 points)

Circle **True** or **False** for each statement below.

True False All if statements must have a corresponding else clause.

True False A while loop can always be rewritten using a for loop.

True False (A && B) || (A || B) is logically the same as (A || B).

True False The C++ compiler always initializes variables to a default value.

More Loops (12 points)

1. Rewrite the following code as a for loop. What is printed?

```
int i = 1;
while (i <= 7)
{
   if (i % 2 == 0)
         cout << i << " ";
   i *= 2;
}
cout << endl;
```

2. Write code to sum the odd integers between 3 and 69 (inclusive) using a **while** loop.

Expressions (12 points)

```
int Num1 = 7;
double Num2 = 4.5;
```

Answer all the following using the variables' initial values above. (LHS = Left Hand Side, RHS = Right Hand Side)

(a)

```
Num1 = (6  +  Num1  *  2 +  1)  /  4.0;
```

LHS: Value:_____ Data Type:_____

RHS: Value:_____ Data Type:_____

(b)

```
Num2 = Num1  /  10  +  9  %  4;
```

LHS: Value:_____ Data Type:_____

RHS: Value:_____ Data Type:_____

(c)

```
Num2 = ('D'-'A')-1;
```

LHS: Value:_____ Data Type:_____

RHS: Value:_____ Data Type: _____

If-Statements (6 points)

1. Given the following code answer the two questions.

```
char choice;
double money;
cout << “Enter the amount of money you have: “;
cin >> money;
cout << "Enter ‘d’ to divide by 2, ‘m’ to multiply by 1.5 or ‘a’ to
add 6.50: ”;
cin >> choice;

//  THE CODE FOR PARTS b and c GOES HERE

if (choice == ‘d’)
   money = money / 2.0;
else if(choice == ‘m’)
   money = money * 1.5;
else if (choice == ‘a’)
   money = money + 6.50;
else
   money = money;
```

a. (2 points) Assume the user enters 73 and then ‘d’. What will be the value of money be after the if statement?

   i. 146.0
   ii. 36.50
   iii. 79.50
   iv. 109.50
   v. None of the above

b. (2 points) Write an if statement that prints out “Valid” if the money entered is not less than 0 or greater than 100.0, inclusive of 0 and 100.0.

c. (2 points) Write an if statement that prints out “Invalid” if the user enters anything except ‘d’, ‘m’, or ‘a’.

Loops (8 points)

(**1**) How many times will the following loop display "Count!"?

```
for (int i = 0; i < 20; i=i+1)
{
   cout<<"Count!"<<endl;
}
```

a) 19
b) 20
c) 21
d) 10
e) None of the above.

(**2**) What is the output of the following code segment?

```
int n = 2;
while (n > 1 &&  n <= 5)
{
   cout << n << "  ";
}
```

a) 2
b) 2 3 4 ... and so on forever
c) 2 3 4 5
d) 2 2 2 … and so on forever
e) None of the above

(**3**) What will the following code display?

```
int x = 0;
for (int count = -1; count > 0; count=count-1)
    cout << x << " ";
x++;
cout << x << " ";
```

(**4**) Consider the following C++ program fragment.

```
 int i, j, count;
 i = 1;
 while (i < 8)
 {
        count = 0;
        for (j = 0; j <= 4; j = j + 2)
        {
             Count=count+1;
        }

        i = i+1;
 }
```

What is the final value of count?

More Loops (12 points)

Rewrite the following code as a for loop. What is printed?

```
int i=1;
while ( i <= 7)
{
    if (i%2 ==1)
        cout << i << " "; i++
}
cout << endl;
```

Write code to sum the even integers between 1 and 81, inclusive, using a for statement.

Write code to print the integers 2 to 27 using a while loop and the counter variable x. Print exactly five integers per line

Arrays and loops (30 points)

Write code that will determine if the elements of an array are sorted in nondecreasing order. Assume the array was declared with int a[100];. Note: you do not have to sort the array.

Assume the array of integers in the previous question. Write code to output the largest and smallest elements of the array.

Write code that calculates and outputs the sum of the integers n, where $0 < n < 100$, which are not divisible by 7.

Questions on Expressions: 12 points

int Num1 = 6;
double Num2 = 2.5;

Answer all the following using the variables' initial values above.
(a)

Num1 = (2 + Num1 * 5 - 4) / 3.0;

Expression: Value:_____ Data Type:_____

Variable: Value:_____ Data Type:_____

(b)

Num2 = Num1 / 10 + 8 % 3;

Expression: Value:_____ Data Type:_____

Variable: Value:_____ Data Type:_____

(c)

Num2 = ('C'-'A')-1;

Expression: Value:_____ Data Type:_____

Variable: Value:_____ Data Type: _____

Questions on If-Statements: 6 points

1. Given the following code answer the two questions.

```
char choice;
float money;
cout << “Enter the amount of money you have: “;
cin >> money;
cout << "Enter ‘d’ to divide by 2, ‘m’ to multiply by 1.5 or ‘a’ to add 6.50: ”;
cin >> choice;

//  THE CODE FOR PARTS b and c GOES HERE
if (choice == ‘d’)
  money = money / 2.0;
else if(choice == ‘m’)
  money = money * 1.5;
else if (choice == ‘a’)
  money = money + 6.50;
else
  money = money;
```

a. (2 points) Assume the user enters 50 and then ‘m’. What will be the value of money be after the if statement?

   i. 25.0.
   ii. 75.0.
   iii. 56.50
   iv. 50.0
   v. None of the above

b. (2 points) Write an if statement that prints out “Invalid” if the money entered is not between 0 and 100.0, inclusive of 0 and 100.0.

c. (2 points) Write an if statement that prints out “Invalid” if the user enters anything except ‘d’, ‘m’, or ‘a’.

Questions on Loops; Output. 8 points

(**1**) How many times will the following loop display "Count!"?

```
for (int i = 0; i < 10; i=i+1)
{
  cout<<"Count!"<<endl;
}
```

a) 5
b) 6
c) 10
d) 11
e) None of the above.

(**2**) What is the output of the following code segment?

```
int n = 1;
while (n >= 5 || n<0)
{
  cout << n << "  ";
  n=n+1;
}
```

a) 1 2 3 4 5
b) 1 1 1 ... and so on forever
c) 1 2 3 … and so on forever
d) 1 2 3 4
e) None of the above

(**3**) What will the following code display?

```
int x = 0;
for (int count = 4; count > 0; count=count-1)
   cout << x << " ";
   x=x+1;
```

a) 4 3 2 1
b) 0 1 2 3
c) 1 2 3 4
d) 0 0 0 0
e) none of the above

(**4**) Consider the following C++ program fragment.

```
int i, j, count;
i = 1;
while (i < 5)
{
    count = 0;
    for (j = 0; j < 6; j = j + 2)
    {
       count=count+1;
    }

    i = i+1;
}
```

What is the final value of count?

a) 4
b) 3
c) 12
d) 0
e) none of the above

Miscellaneous True/False (6 points)

Circle True or False for each statement below

True False (A && B) || (A || B) is logically the same as (A || B)

True False A for loop can always be rewritten using a while loop

True False A while loop always executes its body once

True False If a program compiles, it has no errors

Syntax Errors

[10 points] The following C++ programs have at least one syntax error. For each of the program below, circle the syntax error and explain what is wrong. You can assume that we have correctly included iostream at the top of the program.

```
//-------------------------
int main()
{
        n=0;
}


//-------------------------
int main()
{
        cout >> "Hello, world!" >> endl;
}


//-------------------------
int main()
{
        int n
        n=0;
}


//-------------------------
int main()
{
        char c = "A";
        int n;
}


//-------------------------
int main()
{
        if (0 == 0) {
                cout<<"Hello, mom!\n";
}
```

```
//-------------------------
int main()
{
	int n;
	double n;
	if (0 == 0) {
		cout<<"Hello, dad!\n";
	}
}


//-------------------------
int main()
{
	for (n=0; n<10; n=n+1)
		cout<<"Hello, brother!\n";
}


//-------------------------
int main()
{
	for (int n=0, n<10, n=n+1)
		cout<<"Hello, sister!\n";
}


//-------------------------
int main()
{
	int n;
	cin << n;
	if  (1 <= n <= 10)
		cout<<"Hello, world!\n";
}

//-------------------------
int main()
{
	int n;
	while( (n != 0)  && (n < 16))
		n=n+1;
	}
}
```

Conditionals

Consider the problem of choosing a place to rent. If you have over $2,000 per month you can rent a house to share with 0-3 roommates. If you have over $1,000 per month you can rent an apartment to share with 0-1 roommates. If you have under $1,000 per month you are stuck renting your parent's basement. The following C++ code implements this decision process.

```
float Money = 0.0;
int Roommates = 0;
cin >> Money >> Roommates;

if ( condition 1 )
  cout << "Rent a house and throw big parties\n";
else if ((Money > 1000) && (Roommates == 0))
  cout << "Rent a one bedroom apartment\n";
else if ( condition 2 )
  cout << "Rent a two bedroom apartment\n";
else if (Money < 1000)
  cout << "Rent your parent's basement\n";
else
  cout << "Sorry, no advice\n";
}
```

[5 points] What should **condition 1** be?

a) (Money > 2000)
b) (Money > 2000) && (Roommates != 0)
c) (Roommates >= 0) && (Roommates <= 3) && (Money > 2000)
d) (Money > 2000) && (0 <= Roommates <= 3)
e) None of the above.

[5 points] What should **condition 2** be?

a) (Money < 2000) && (Roommates < 2)
b) (0 <= Roommates <= 1)
c) (Money * Roommates == 1000)
d) (Money > 1000) && (Roommates == 1)
e) None of the above.

Conditionals

The nested conditional statement shown below has been written by an inexperienced C/C++ programmer. The behavior of the statement is not correctly represented by the formatting.

```
if (n < 12)
     if (n > 0)
          cout << "The number is positive." << endl;
else
     cout << "The number is _______________." << endl;
```

a. **[6 points]** What is the output of the statement if the variable `n` has the value 7? If `n` has the value 15? If `n` has the value –3?

b. **[2 points]** Correct the syntax of the statement so that the *logic* of the corrected statement corresponds to the *formatting* of the original statement. Also, replace the blank with an appropriate word or phrase.

c. **[2 points]** Correct the formatting of the (original) statement so that the new format reflects the logical behavior of the original statement. Also, replace the blank with an appropriate word or phrase.

Topic related to project and lab

[10 points] Consider the following code that converts letter grade (A,B,C,D,F) to score for GPA calculation, where grade is of type char and contains the letter grade and scores is of type int.

```
if (grade == 'D') {
   scores = 1;
} else if (grade == 'B'){
   scores = 3;
// missing code
} else if (grade == 'F'){
   scores = 0;
} else
   cout << "invalid grade value " << grade;
```

Which of the following could replace the missing code so that the code fragment will work as intended?

A. Code fragment

```
} else if (grade == 'A'){
   scores = 4;
```

B. Code fragment

```
} else if (grade == 'A'){
   scores = 4;
} else if (grade == 'C'){
   scores = 2;
```

C. Code fragment

```
} else if (grade == 'C'){
   scores = 4;
} else if (grade == 'A'){
   scores = 2;
```

D. Code fragment

```
else if (grade == 'A'){
   scores = 4;
} else if (grade == 'C'){
   scores = 2;
}
```

Topic related to project and lab

[10 points] Consider the following code fragment that intends to compute the sum the integers that the user enters. Some pieces are missing as indicated by the underlines. Fill in the missing pieces so that the code will work as intended.

```
int n,x,sum;
cout << “Enter how many numbers that you want to sum\n”
cin >> n;
_________________
for(_____________; k < n && n > 0; k++){
   cout << “Enter the next number to be summed: “;
   cin >> x;

   ____________________________________;
}
cout <<“The sum of the “ << n <<“values that you entered is
“ << sum;
```

Loops

[5 points] Rewrite the following as a <u>for loop</u> that performs the same task.

```
int Num = 1;
int Sum = 0;
while ((Sum < 100))
{
  Sum = Sum + Num;
  Num = Num + 2;
  cout << "Sum: " << Sum << endl;
}
```

[5 points] Rewrite the following as a <u>while loop</u> that performs the same task.

```
char Reply;
cout << "Want to quit (y/n)? ";
cin >> Reply;
for (int i= 0; (Reply != 'y') && (Reply != 'n');  i=i+1  ){
        cout << "Want to quit (y/n)? ";
        cin >> Reply;
}
```

[5 points] Rewrite the following as a while loop that produces the same output.

```
int Value;
for (Value = 0; Value < 42; Value=Value-1)
{
  cout << "Value: " << Value << endl;
  Value = Value + 3;
}
```

[5 points] What does the following program print out?

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

int main()
{
  int A, B;
  for (A = 0; A <= 4; A=A+1)
  {
    cout << A << ":";
    B = A + 1;
    while(B < 5)
    {
      cout << "*";
      B = B + 1;
    }
    cout << "\n";
  }
  cout << "Final value of A: " << A << "\n";
  cout << "Final value of B: " << B << "\n";
  return(0);
}
```

Arrays and loops

[10 points] Determine the output of the following C++ program when it is executed.

```
int a[5];
a[4] = 20;
for (int i=4; i>0; i=i-1;)
     a[i-1] = a[i]- i;
for (int i=0; i<5; i++;)
     cout << a[i] << endl;
```

Understanding loop

[10 points] Consider the following code fragment, x has type int.

```
int sum;
x = x/2;
for (sum = 1; x > 0; x = x/2)
   sum = sum + 1;
```

Which of the following will calculate the same value of sum as the fragment above?

A. Code fragment

```
int sum = 0;
x = x/2;
while (x => 0) {
   sum = sum + 1;
   x = x/2;
}
```

B. Code fragment

```
int sum = 1;
x = x/2;
while (x => 0) {
   sum = sum + 1;
   x = x/2;
}
```

C. Code fragment

```
int sum = 0;
x = x/2;
while (x > 0) {
   sum = sum + 1;
   x = x/2;
}
sum = sum + 1;
```

D. Code fragment

```
int sum = 1;
x = x/2;
while (x > 0) {
   sum = sum + 1;
   x = x/2;
}
```

Conditionals

Consider the problem of choosing a place to rent. If you have over $2,000 per month you can rent a house to share with 0-3 roommates. If you have over $1,000 per month you can rent an apartment to share with 0-1 roommates. If you have under $1,000 per month you are stuck renting your parent's basement. The following C++ code implements this decision process.

```
float Money = 0.0;
int Roommates = 0;
cin >> Money >> Roommates;

if ( condition 1 )
  cout << "Rent a house and throw big parties\n";
else if ((Money > 1000) && (Roommates == 0))
  cout << "Rent a one bedroom apartment\n";
else if ( condition 2 )
  cout << "Rent a two bedroom apartment\n";
else if (Money < 1000)
  cout << "Rent your parent's basement\n";
else
  cout << "Sorry, no advice\n";
}
```

[5 points] What should **condition 1** be?

f) (Money > 2000)
g) (Money > 2000) && (Roommates != 0)
h) (Roommates >= 0) && (Roommates <= 3) && (Money > 2000)
i) (Money > 2000) && (0 <= Roommates <= 3)
j) None of the above.

[5 points] What should **condition 2** be?

f) (Money < 2000) && (Roommates < 2)
g) (0 <= Roommates <= 1)
h) (Money * Roommates == 1000)
i) (Money > 1000) && (Roommates == 1)
j) None of the above.

Conditionals

When you buy a new car, the dealer's profit margin is often based on the list price of the car. The bigger the sticker price, the more markup you pay. The following C++ code implements a crude approximation of this pricing scheme. There is a 20% profit on cars over $30,000, a 10% profit on cars over $20,000 and only $500 profit on cars less than $20,000 (if you can find one on the lot).

```
float Profit1 = 0.2;
float Profit2 = 0.1;
float ListPrice = 0.0;
float SalePrice = 0.0;

cin >> ListPrice;
if (ListPrice >= 30000.0)
  SalePrice = ListPrice + ListPrice * Profit1;
else if (ListPrice >= 20000.0)
  SalePrice = ListPrice + ListPrice * Profit2;
else
  SalePrice = ListPrice + 500;
cout << "$" << fixed << setprecision(2) << SalePrice << endl;
```

[5 points] Assume the user has entered 20000.0 as the list price, what will this program print out?

a) $20000
b) $20000.00
c) $22000.00
d) $22222.22
e) $24000.00

[5 points] Assume the user has entered 40000.0 as the list price, what will this program print out?

a) $40000
b) $44000.00
c) $44444.44
d) $48000.00
e) $40500.00

Loops

[5 points] Rewrite the following as a for loop that performs the same task.

```
int Num = 1;
int Sum = 0;
while ((Sum < 100))
{
  Sum = Sum + Num;
  Num = Num + 2;
  cout << "Sum: " << Sum << endl;
}
```

[5 points] Rewrite the following as a while loop that performs the same task.

```
char Reply;
do
{
  cout << "Want to quit (y/n)? ";
  cin >> Reply;

} while ((Reply != 'y') && (Reply != 'n'));
```

[5 points] Rewrite the following as a while loop that produces the same output.

```
int Value;
for (Value = 0; Value < 42; Value--)
{
  cout << "Value: " << Value << endl;
  Value = Value + 3;
}
```

[5 points] What does the following program print out?

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

int main()
{
  int A, B;
  for (A = 0; A <= 4; A++)
  {
    cout << A << ":";
    B = A + 1;
    while(B < 5)
    {
      cout << "*";
      B = B + 1;
    }
    cout << "\n";
  }
  cout << "Final value of A: " << A << "\n";
  cout << "Final value of B: " << B << "\n";
  return(0);
}
```