University of Calgary
Faculty of Engineering
ENGG 233 – Computing for Engineers I
Midterm Examination for 2010 Winter Session
18:00 to 20:00, Tuesday, 16 March 2010

This midterm exam counts for 40% of your total course grade.

NAME (PRINT): _____
Last Name Given Name(s)

LECTURE SECTION (circle):
L01 (Nassar – TuTh @ 11:00) L02 (Petovello – TuTh @ 9:30)
L03 (Moussavi– MoWeFr @ 10:00) L04 (Far – MoWeFr @ 13:00)
L05 (Moussavi – TuTh @ 12:30) L06 (Liang – MoWeFr @ 15:00)

SIGNATURE: _____

Be sure to put your name and U of C ID number on the bubble sheet, and FILL IN THE APPROPRIATE BUBBLES. Failing to fill in the bubbles or filling them incorrectly may mean your bubble sheet cannot be referenced back to you and you may thus forfeit the corresponding marks.

Instructions

- To prevent distracting your fellow students, you may not leave the exam room during the last ten minutes of the exam.
- **No electronic devices <u>of any kind</u> may be used during this exam.**
- **<u>This is a closed-book test.</u>** You must not refer to books or notes during the test.
- Write legibly! What cannot be read will not be marked.
- If you remove the staple from the paper, write your name at the top of every loose page.
- Attempt all questions.
- Get the easiest marks first! If you find a particular question difficult, return to it after completing easier questions.
- If you write any rough work on the question paper, cross it out with a large "X" and write "Rough work" beside it. Rough work will not be marked.
- Follow the directions for each problem carefully.

Some Predefined Functions from cmath and stdlib

Prototype	Library	Description	Example	Value of Example
`int abs(int i);`	cstdlib	absolute value	abs(-2)	2
`double fabs(double x);`	cmath	absolute value	fabs(-2.4)	2.400000000
`double pow(double x, double y);`	cmath	power x^y	pow(2,3) $\Rightarrow 2^3$ pow(4,0.5) $\Rightarrow \sqrt{4}$	8.0000 2.0000
`double sqrt(double x);`	cmath	square root	sqrt (3)	1.732050808
`double exp(double x);`	cmath	exponential e^x	exp(1)	2.718281828 $e \approx 2.718281828$
`double log(double x);`	cmath	base e logarithm	log (2.7183) [$2.7183 \approx e$]	1.000006685 [$1.000006685 \approx 1$]
`double log10(double x);`	cmath	base 10 logarithm	log10(100)	2.0000
`double sin(double x);`	cmath	sine of <u>radian</u> angle	sin(0.7854) [$0.7854 \approx \pi/4$]	0.707108080 [$0.7071 \approx 1/\sqrt{2}$]
`double cos(double x);`	cmath	cosine of <u>radian</u> angle	cos(3.14) [$3.14 \approx \pi$]	-0.999998732 [$-0.999998732 \approx -1$]

Grading

Question	1	2	3	4	5	Total
Marks	/ 10	/ 10	/ 5	/ 15	/ 20	/ 60

U of C ID Number:	

SECTION 1 [*10 marks*] MULTIPLE CHOICE

For each of the following, select the most appropriate answer. Indicate your answer on the bubble sheet provided.

1) Which of the following is not a keyword in C?
   a) `true`
   b) `if`
   c) `integer`
   d) `switch`

2) Which of the following is an invalid logical expression? Assume that all variables are `int`.
   a) `a = b || c <= d`
   b) `a >= b && x != 5`
   c) `10.5 => x`
   d) None of the above
   e) All of the above

3) Which of the following is a correct variable declaration?
   a) `int 6pack`
   b) `double can volume;`
   c) `int return;`
   d) `bool _is_true;`

4) What will be the value inside the variables x and y after the given set of assignments?

```
int x = 20;
int y = 10;
x += --y * 2;
```

   a) x = 38, y = 9
   b) x = 40, y = 9
   c) x = 18, y = 9
   d) x = 20, y = 9
   e) None of the above

5) How many times does the following loop run?

```
int i = 2;
int j = 4;
do
{
   i++;
   if (i % j == 0)
   {
      i--;
   }
}
while (i >= 1);
```

   a) 1 times
   b) 2 times
   c) 3 times
   d) This is an infinite loop

6) Consider the code below:

```
int i, j;
for (i=6 ; i>4 ; i--)
     for (j=6 ; j>3 ; j--)
         cout<<"*";
```

   How many asterisks does this code print in total?

   a) 5
   b) 6
   c) 9
   d) 18

7) What is the value of i after executing the given code segment?

```
int i = 10 ;
while (i>0){
   cout<<"I love ENGG233" ;
   i-=2 ;
}
```

a) 0
b) 2
c) -2
d) I love ENGG233

8) What is the value of `i` after executing the given code segment?

```
int i = 46/10 ;
```

a) 4.6
b) 4
c) 5
d) None of the above

9) Which set of the two function prototypes represents overloaded functions?

a)
```
int weird();
double weird();
```
b)
```
int weird(int b, int a);
double weird(int a, int b);
```
c)
```
char weird();
int weird(int a);
```
d) All of the above
e) None of the above

10) Which of the following logical expression returns true when x is less than 20 and greater than 0?

a) `20>x>0`
b) `x<20 && x>0`
c) `(20>x>0)`
d) None of the above

SECTION 2 [Total: 10 *marks*] OUTPUT DETERMINATION

For each of the following, identify the most appropriate output. Indicate your answer on the bubble sheet provided.

11) What is the output of the following code segment? [This should be question 11, since it will be written on the bubble sheet. Also, the number has a different font.]

```
int main()
{
   int var = 2;
   if (var == 2)
      {
        int var = 10;
        cout << var <<" ";
      }
   cout<<var;
   return 0;
}
```

a) 2 2
b) 10 10
c) 2 10
d) 10 2

12) What is the output of the following code segment?

```
int main()
{
  cout << 100 << 2 + 2;
  return 0;
}
```

a) 100 4
b) 100 2+2
c) 104
d) 1004

13) What is the output of the following code snippet?

```
int wow(int a, int b)
{
   return a*b;
}
int main()
{
   cout << wow(wow(4, 2), wow(3,2)) << endl;
   return 0;
}
```

a) 12
b) 48
c) 144
d) There is a syntax error in the code

14) What is the output of the following code snippet?

```
int main()
{
   double a = 123.456789 ;
   cout<<setiosflags(ios::fixed)<<setprecision(3)<<a ;
   return 0;
}
```

a) 123.45
b) 123.456
c) 123.457
d) 123

15) What is the output of the following code snippet?

```
int main()
{
   char a = '1';
   if (a <= 'b')
   {
      cout << "I";
   }
   if (a <= 'B')
   {
      cout << "LOVE";
   }
   else if (a <= '2')
   {
      cout << "ENGG233";
   }
   return 0;
}
```

a) I
b) LOVE
c) ILOVE
d) ILOVEENGG233

16) What is the output of the following code segment?

```
int i = 10;
while(0)
{
   cout << i << " ";
   i = i+2;
}
cout<<"*";
```

a) *
b) 0 2 4 6 8 10 12 14 …. (infinite loop)
c) 10 12 14 16 18 …. (infinite loop)
d) There is a syntax error in the code.

17) What is the output of the following switch statement? [The question should all be on one page. Leave the bottom of the page blank, or re-order questions, if necessary.]

```
int main()
{
      enum color {RED,BLACK,BLUE,YELLOW};
      color car_color = BLACK ;
      int a = 2;
      switch(car_color)
      {
            case 1: a++;
                  break;
            case 2: a--;
                  break;
            case 3: --a;
                  break;
            case 4: a*=3;
                  break;
            default: a=a*5;
      };
      cout<<a<<endl;
}
```

a) 3
b) 2
c) 9
d) 15

18) Which of the following options represents the output of the given code segment?

```
void weird(int x, int& y, int z)
{
      y = x + 5;
      z = x - 5;
}
int main()
{
      int a = 5;
      int b = 0;
      int c = 0;
      weird(a, b, c);
      cout << b << " " << c;
      return 0;
}
```

a) 10 -5
b) 0 0
c) 10 0
d) None of the above

19) What is the output of the following code segment?

```
double a = 6;
int b = 10;

cout<<static_cast<int>((10.6+a)/b);
```

a) 0
b) 1
c) 2
d) 1.66

20) Which of the following options represents the output of the given code segment?

```
int a =4;
string temp;
temp = (1/a != a/2)?"ENGG":"233";
cout<<temp;
```

a) ENGG
b) 233
c) ENGG233
d) None of the above

SECTION 3 [Total: 5 *marks*] SCOPE AND LIFE TIME

In the table below, indicate with an X which variables and functions are available for each of the code positions indicated by (1) to (5). Available means that these variables and functions could be used at these code positions without adding compiler errors.

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

string no = "NO!!!!!";
void chorus(string, int);

//(1)
int main()
{
      int holmer = 10, bart = 2;
      string number;
      cin>>number;
      string lyrics = number + " bottles of beer on the wall";

      //(2)

      if(holmer>1){
            chorus(lyrics, holmer);
            //(3)
      }
      return 0;
}

void chorus(string a, int b)
{
      if(b>1)
            for (int i=0; i<b;i++){
                  cout<<a<<endl;
                  //(4)
            }
      else
      {
            string warning_message = no;
            cout<<warning_message;
            //(5)
      }
      return;
}
```

		(1)	(2)	(3)	(4)	(5)
Functions	main					
	chorus					
Variables	no					
	holmer					
	bart					
	number					
	lyrics					
	a					
	b					
	i					
	warning_message					

SECTION 4 [Total: 15 *marks*] FUNCTIONS

Function Interface (4 marks)

In the boxes below, write a **function prototype only** based on the description of what the function is expected to accomplish. **Do not write the function definition**.

1. (**1 mark**) A function to read an integer into a variable that is passed in as an argument. The function should return true if successful and false otherwise.

2. (**2 marks**) A function to print a student's name, ID number and GPA to the screen. All three outputs should be passed in as arguments.

3. (**1 mark**) An overloaded version of the function **`int Add( int, int )`**, which adds two numbers and returns the result, but that works with variables of type double.

Function Implementation (11 marks)

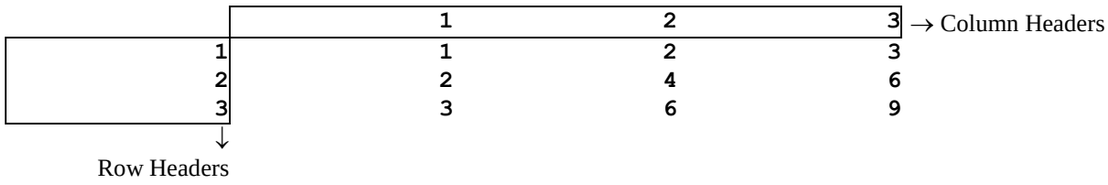
Implement the following functions in the boxes provided below. Assume all necessary #include statements and function prototypes appear above the function implementations.

1. (**5 marks**) Write a C++ function to prompt the user for a number in the range of 'minValue' to 'maxValue' (inclusive) and return it. If the user inputs a value outside this range, the function should output "*Invalid input. Please try again.*" and try to read another value from the keyboard. If the maximum value is less than the minimum value, the function should immediately return a value of zero.

```
int PromptForNumberInRange( int minValue, int maxValue )
{

}
```

2. (**6 marks**) Write a C++ function to print a square multiplication table of a user-specified size (N by N), with the table running from 1 to N. The table should print row and column "headers" to identify the values used in the multiplication. For example, if the input is three (i.e., $N = 3$), the following should be displayed:



The width of each column depends on N. It means each column's field size must be calculated based on the following equation:

column_width = $\log_{10}\left(N^2\right) + 2$.

In the above, you do **not** need to output any lines, nor the text "Column Headers" or "Row Headers" (these are included for clarity only).

```
void DisplayMultiplicationTable( int N )
{




}
```

SECTION 5 [Total: 20 *marks*] PROGRAM IMPLEMENTATIONS

For this question, you will write a program along with supporting functions to simulate the casino game of craps. Craps is a game played with two six-sided dice (sides are numbered 1 to 6). The game is described in more detail below.

How to Play Craps

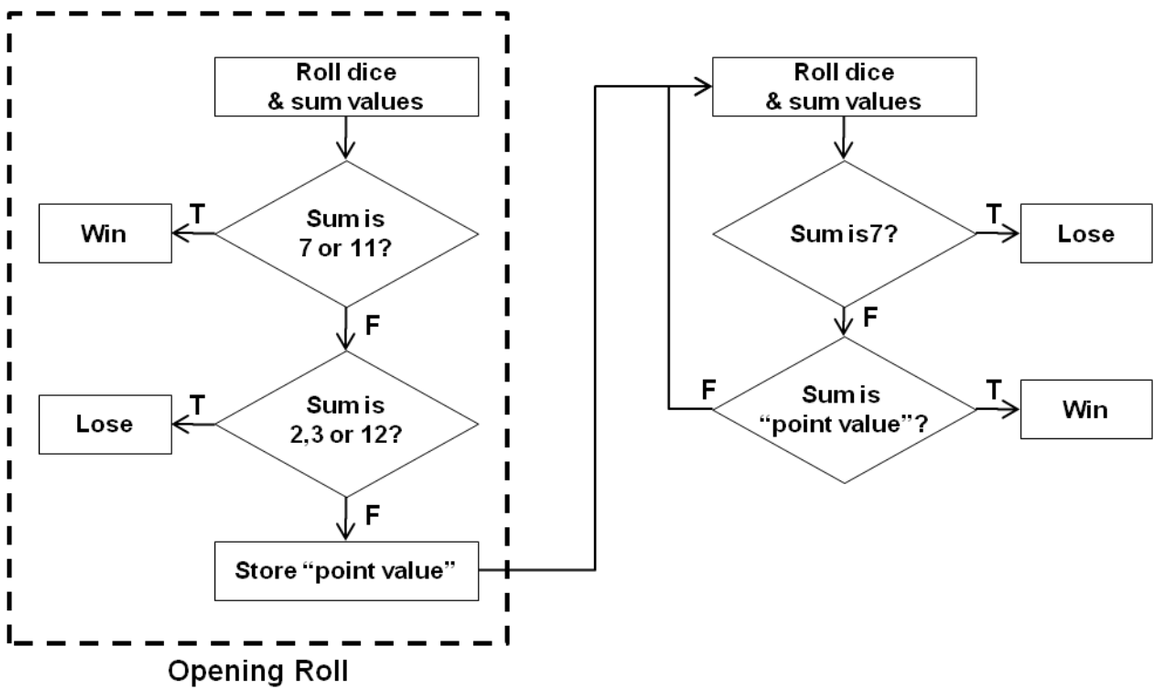
The basic game starts with an *opening roll*, in which the player throws both dice. The outcome of the opening roll can be one of the following:

- If the sum of the two dice is 7 or 11, the player wins and the game is over.
- If the total is 2, 3 or 12, the player loses and the game is over.
- Any other total (i.e., not 2, 3, 7, 11 or 12) becomes the **"point value".**

If the game is not over based on the opening roll, the player continues to roll the dice (in pairs) until one of following things happens:

- If the dice total is equal to the "**point value**" , the player wins and the game is over.
- If the dice total 7, the player loses and the game is over.

To further explain how the game is played, a *simplified* flowchart is given below. It does not contain all of the steps needed to complete the program (e.g., no output).



Below are some sample outputs from the program (one run per column to save space). In games #1 and #2, the game ends after the opening roll. In game #3, the player wins because they rolled the "point value" (5, in that example) before rolling a 7. In game #4, the player loses because they rolled a 7 before they rolled the "point value" (8, in that example).

Game #1	**Game #2**	**Game #3**	**Game #4**
Player rolls 2 + 5 = 7	Player rolls 1 + 1 = 2	Player rolls 4 + 1 = 5	Player rolls 3 + 5 = 8
Player wins!	Player loses!	Point value is 5	Point value is 8
		Player rolls 4 + 6 = 10	Player rolls 6 + 6 = 12
		Player rolls 1 + 1 = 2	Player rolls 6 + 1 = 7
		Player rolls 3 + 2 = 5	Player loses!
		Player wins!	

Program Details

In total, the program will consist of three functions plus the main function. Of the three functions, assume that one of them has already been implemented in a library that has been provided to you (i.e., you do **not** need to implement this function yourself). The function has the following prototype and documentation.

```
int RollOneDie();
// Function to return a random number to simulate the rolling of a die.
//
// REQUIRES:
//    No requirements.
//
// PROMISES:
//    Returns a random value between 1 and 6 (both inclusive) to simulate the
//    rolling of a die.
```

You are responsible for writing the remaining two functions plus the main function. Detailed information about the two functions and main is given below. Students are encouraged to read **all** parts of the question **before writing any code** in order to get a good idea of how everything works together.

Feel free to remove this page. It may be convenient to see the questions and answer sheets on pages 11-13.

There is no need to return this page with the exam paper.

What to Do

In the boxes below, write the implementations for each of the above functions. Some notes:

- You do **not** need to add the `#include` statements at the top of your program.
- You need to decide the return type as well as the number and type of function arguments yourself.

A. (**3 marks**) Write the `ThrowDice` function. The function should call the `RollOneDie` function once for each die and output a message saying "*Player rolls X + Y = Z*" where X is the value of the first die, Y is the value of the second die and Z is the total of X and Y. The function should return the sum of the two dice (Z, in the above example).

B. (**9 marks**) Write the **`OpeningRoll`** function that receives an integer argument that represents the "point value". The function should

- Call the **`ThrowDice`** function and store the sum of the two dice.
- If the sum of the dice totals 7 or 11, the function should output "*Player wins!*".
- If the sum of the dice totals 2, 3, or 12, the function should output "*Player loses!*".
- If the player wins or loses, the function should return **`true`**, indicating the game is over, and the value of the point value should be set to -1.
- For any other total of the dice, the function should output "*Point value is Z*" and return **`false`** (with Z is the sum of the dice).

C. (**8 marks**) Write the **`main`** function. The function should

- Call the **`OpeningRoll`** function. If the game is over based on the opening roll, the function should end.
- If the game is not over after the opening roll, a **do-while** loop should be used to simulate each additional roll of dice. The loop should terminate when the dice total either 7 or the "point value".
- The function should then output "*Player wins!*" if the dice total the "point value" and it should output "*Player loses!*" if the dice total 7.
- You do **not** need to add the **`#include`** statements at the top of your program.