Homework Assignment #0

1. C++ syntax reviews. Please read your textbook, and answer the following questions. Here you are asked to review the syntax (i.e., grammar rules) and the meaning (i.e., the effect) of C++ expressions and statements. Please answer all questions based upon the example given below:

(a) Write down the syntax and meaning of assignment statement.
   Answer: The syntax of assignment statement is:
   \[ \text{variable} = \text{expressions}; \]
   When executed, the expression on the right-hand side is evaluated, and its value is assigned to the variable on the left-hand side.

(b) Write down the syntax and meaning of if-else statement.
   Answer: The syntax of if-else statement is:
   \[
   \begin{align*}
   &\text{if (condition)} \\
   &\quad \text{yes_statement;} \\
   &\text{else} \\
   &\quad \text{no_statement;} \\
   &\text{next_statement;}
   \end{align*}
   \]
   When executed, the condition is evaluated first, if its value is true, yes_statement is executed; if its value is false, no_statement is executed.

(c) Write down the syntax and meaning of if statement.

(d) Write down the syntax and meaning of for statement. Based on this, write a for loop to print out 100 of character $ on the terminal.

(e) Write down the syntax and meaning of while statement.

(f) Write down the syntax and meaning of do-while statement. Based on this, write a code segment that reads from keyboard the month value, and repeats until a valid month value is entered.

(g) What is dangling else problem? How is it solved? Based on this, find out the output generated by the following code segment:
   \[
   \begin{align*}
   &\text{int x=1, y=2;} \\
   &\text{if (x==3)} \\
   &\quad \text{cout <<" *****\n";} \\
   &\text{if (y==3)} \\
   &\quad \text{cout <<" @@@@\n";} \\
   &\text{else} \\
   &\quad \text{cout <<"!!!!\n";} \\
   \end{align*}
   \]

(h) What is empty statement (or, null statement)? Does the following code segment contain syntax error? If not, what’s the effect?
   \[
   \begin{align*}
   &\text{int a;} \\
   &\text{cin >> a;} \\
   &\text{if (a<0);} \\
   &\text{else} \\
   &\quad \text{cout <<"a is not negative\n";} \\
   \end{align*}
   \]
(i) Based upon the syntax rule of for loop and empty statement as you have reviewed, find out the output of the following code segment:

```cpp
int sum=0;
int i;

for (i=0;i++;i<4);
sum = sum+i;

cout <<"sum=" << sum << endl;
```

2 Review C++ operators and precedence rules.

(a) List all arithmetic operators in C++, such as + for addition.
(b) List all relational operators (i.e., comparison operators) in C++, such as == for "is equal" testing.
(c) List all boolean operators, such as &&.
(d) What is the purpose of precedence rules? For all above operators, list them from high precedence to low precedence.
(e) What is the purpose of association rules? What’s the association rule for the following operators respectively?

   +, *, =, ==

(f) What are the values of the following expressions?

1. \(1 > 0 || 5 > 4 \&\& 3 == 9\)
2. \(a = 10 * 2\)
3. \((a = 10) * 2\)
4. \(1 < x < 10\), where int variable x’s value is 20

3 Review function calls, and the difference between pass-by-value and pass-by-reference parameters. Trace the execution of the following code segment, and identify the outputs generated by them.

(a) Identify the output of the following program as it is.

```cpp
void CheckEvenNumber (int a, bool IsEven)
{
    if (a % 2 ==0)
        IsEven = true;
    else
        IsEven = false;
}

int main()
{
    int input=124;
    bool even=false;

    CheckEvenNumber (input, even);
    if (even)
        cout <<"it’s even\n";
    else
        cout <<"it’s odd\n";
}
```
void factor (int & number)
{
    if (number==1)
        return;
    if (number==2)
        cout << number << " " << endl;
    else
    {
        for (int f=2; f<=number; f++)
        {
            if (number % f == 0)
            {
                number = number / f;
                cout << f << " " << endl;
                factor (number);
                break;
            }
        }
    }
}

int main()
{
    int num=27;
    factor (num);
}

bool SameCharArray (char array_a[], char array_b[], int size)
{
}

This function reads from keyboard the length and height of a rectangle
@param length: to store its length
@param height: to store its height
Postcondition: valid values of rectangle's dimension have been
stored in the two parameters, i.e., both needs to be positive values
in another word, if negative values are entered, the function should
report errors and prompt the user to reenter
*/
void ReadRectangleDimension (int & length, int & height)
{
}