

SHEET 2 SOLUTIONS

SHEHAB MAHMOUD SALAH | 2100320

GENERAL NOTE: to **copy** code **from** this PDF document, **copy each block of code** separately to **not** lose the code's formatting, alternatively you can find all the **source code** to **PROGRAMMING EXERCISES** on my **GitHub**: <https://bit.ly/CSE131Sheets> happy compiling 😊

(A) Review Questions

1. A) Wrong syntax, the correct statement would be:

```
if (quarters > 0){  
    cout << quarters << "quarters";  
}
```

B) Wrong syntax, the correct statement would be:

```
if (1+x > pow(x , sqrt(2))) {  
    y = y+x;  
}
```

C) The correct code would be:

```
if (x == 1) {  
    y++;  
}  
else if (x == 2) {  
    y = y + 2;  
}
```

D) The correct code would be:

```
if (x == 0 && y == 0) {  
    cout << Point(0,0);  
}
```

E) The correct code would be:

```
if (1 <= x && x <= 10) {  
    cout << "Enter y: ";  
    cin >> y;  
}
```

F) The correct code would be:

```
if (s != "nick" && s != "penn" && s != "dime" && s != "quar") cout  
<< "Input error!";
```

Explanation: The logical operator that should be used is the AND operator, because using the OR operator means the condition will always be true regardless of what "s" is, and you will always get "Input error!" as an output.

G) `if (input == "N" || "NO") return 0;`

H) `cin >> x; if(!cin.fail()) y = y + x;`

```
l) language = "English";  
if (country == "USA") {  
    if (state == "PR") language = "Spanish";
```

```

    }
    else if (country = "China") language = "Chinese";

```

2. A. Tom, Dick - **Dick comes first**
- B. Tom, Tomato - **Tom comes first**
- C. Car manufacturer, carburetor - **Car manufacturer comes first**
- D. Harry, hairy - **Harry comes first**
- E. C++, Car - **C++ comes first**
- F. Car, Carl - **Car comes first**

3.
 - A) **15.9** The stream will read 2 characters (15) and stop at the decimal point. **The stream will not fail.**
 - B) **15 9** The stream will read 2 characters (15) and stop at the space. **The stream will not fail.**
 - C) **+159** The stream will read 4 characters (+159) and stop at the end of line. **The stream will not fail.**
 - D) **-15A9** The stream will read 3 characters (-15) and stop at A. **The stream will not fail.**
 - E) **Fifteen** **The stream will fail immediately as F is not a digit or a sign.**
 - F) **-Fifteen** **The stream will fail immediately as F is not a digit after -.**
 - G) **+ 15** **The stream will fail immediately as there is no digit after +.**
 - H) **1.5E3** The stream will read 1 character (1) and stop at the decimal point. **The stream will not fail.**
 - I) **+1+5** The stream will read 2 characters (+1) and stop at +. **The stream will not fail.**

4. An if/else/else statement is a way of branching the execution flow based on different conditions. For example:

```

if (x > 0) {
    cout << "x is positive" << endl;
}
else if (x < 0) {
    cout << "x is negative" << endl;
}
else {
    cout << "x is zero" << endl;
}

```

This code will print different messages depending on the value of x. The else if clause is executed only if the previous if condition is false. The else clause is executed only if all the previous conditions are false.

A nested if statement is an if statement that is inside another if statement. For example:

```

if (y > 10) {
    cout << "y is greater than 10" << endl;
    if (y % 2 == 0) {
        cout << "y is also even" << endl;
    }
}

```

This code will print two messages only if y is greater than 10 and even. The inner if condition is evaluated only if the outer if condition is true.

Nested if statements can be equivalent to single if statements with a combined condition using logical operators like && (and) or || (or). For example, the above code can be rewritten as:

```

if (y > 10 && y % 2 == 0) {

```

```
    cout << "y is greater than 10 and even" << endl;
}
```

5. Here is an example of an if/else/else statement where the order of the tests does not matter:

```
if (x > 0 && y > 0){
    cout << x + y << endl;
} else {
    cout << x - y << endl;
}
```

In this example, the if statement will always print the sum of x and y, regardless of the order of the tests.

Here is an example of an if/else/else statement where the order of the tests matters:

```
if (x > y) {
    cout << x << endl;
} else {
    cout << y << endl;
}
```

In this example, the if statement will only print x if x is greater than y, and will only print y if y is greater than x. The order of the tests matters because the if statement will only check the first test that is true.

6.

p	q	r	(p && q) !r	!(p && (q !r))
0	0	0	1	1
0	0	1	0	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	0

7. To formulate the condition "**x is positive**", you can use the following code:

```
if (x > 0) {
    // Code to be executed if x is positive
}
```

To formulate the condition "**x is zero or negative**", you can use the following code:

```
if (x <= 0) {
    // Code to be executed if x is zero or negative
}
```

To formulate the condition "**x is at least 8**", you can use the following code:

```
if (x >= 8) {
```

```
// Code to be executed if x is at least 8  
}
```

To formulate the condition "**x is less than 8**", you can use the following code:

```
if (x < 8) {  
    // Code to be executed if x is less than 8  
}
```

To formulate the condition "**x and y are both zero**", you can use the following code:

```
if (x == 0 && y == 0) {  
    // Code to be executed if x and y are both zero  
}
```

8. The difference between the two code segments is that the second one has an else if statement. **This means that if x is positive, s will be incremented, and if y is positive, s will also be incremented.** However, if x is negative, only y will be incremented.

In the first code segment, **s will only be incremented if x is positive.** If y is negative, s will remain at 0.

9. **An infinite loop is a sequence of instructions that repeats itself indefinitely.** This can be done by using a while loop or a for loop.

To terminate a program that executes an infinite loop, you can use the CTRL+C keyboard shortcut. This will send a SIGINT signal to the program, which will cause it to stop executing.

You can also use the Task Manager to terminate a program that executes an infinite loop. To do this, open the Task Manager and find the program that is executing the infinite loop. Then, right-click on the program and select "End Task".

10. A. The loop will execute 10 times, starting from i = 1 and ending at i = 10.

B. The loop will execute 9 times, starting from i = 0 and ending at i = 9.

C. The loop will execute 9 times, starting from i = 10 and ending at i = 1.

D. The loop will execute 21 times, starting from i = -10 and ending at i = 10.

E. The loop is an infinite loop.

F. The loop will execute 11 times, starting from i = -10 and ending at i = 10.

G. The loop will execute 7 times, starting from i = -10 and ending at i = 7.

```
11. int s{};  
int i = 1;  
while (i <= 10) {  
    s = s + i;  
    i++; }  

```

12.

```
int n;  
cin >> n;  
double x = 0;  
double s = 0;  
while (s > 0.01)  
{  
    s = 1.0 / (1 + n * n);  
    n++;  
    x = x + s;  
}
```

The only difference is that the while loop checks the condition at the top of the loop, while the do/while loop checks the condition at the bottom of the loop. This means that the do/while loop will always execute the loop body at least once, even if the condition is false.

13. A. **s = 10, n = 2**

B. **s = 11, n = 5**

C. **s = 211, n = 21**

14.

A. s = 1, n = 1 => s = 2
-> n = 2 => s = 4
-> n = 3 => s = 7
-> n = 4 => s = 11
-> n = 5 => s = 16

∴ Printed values are: 2, 4, 7, 11, 16

B. s = 1, n = 1
-> n = 3, s = 4 => 4
-> n = 5, s = 9 => 9
-> n = 7, s = 16 => 16

∴ Printed values are: 4, 9, 16

C. s = 1, n = 1
-> s = 2, n = 2, n = 3
-> s = 5, n = 4, n = 5
-> s = 10, n = 6, n = 7

∴ Printed values are: 10, 7

15. A. n = 1, i = 2 => n = 3 -> i = 3
=> n = 6 -> i = 4
=> n = 10 -> i = 5 (break.)

n = 0.5, i = 2 => n = 0.75 -> i = 3
=> n = 0.583 -> i = 4
=> n = 0.396 -> i = 5
=> n = 0.279 -> i = 6

B. x = 1, y = 1, i = 0 -> y = 0.5, x = 1.5 -> i = 1 -> y = 0.25, x = 1.75 -> i = 2 -> y = 0.125, x = 1.875 -> i = 3

C. 0

(B) Programming Exercises

```
1. #include <iostream>
   #include <cmath>
   using namespace std;

   int main()
   {
       // Declare variables
       double a, b, c;
       double discriminant, root1, root2;

       // Read in coefficients
       cout << "Enter coefficients a, b and c: ";
       cin >> a >> b >> c;

       // Calculate discriminant
       discriminant = b * b - 4 * a * c;

       // Check if discriminant is negative
       if (discriminant < 0)
       {
           // Display message
           cout << "There are no solutions." << endl;
       }
       else
       {
           // Calculate roots using quadratic formula
           root1 = (-b + sqrt(discriminant)) / (2 * a);
           root2 = (-b - sqrt(discriminant)) / (2 * a);

           // Display roots
           cout << "The solutions are " << root1 << " and " << root2 <<
endl;
       }

       return 0;
   }
```

2.

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

int main() {
    string input;
    cout << "Enter a card notation: ";
    cin >> input;
    if (toupper(input[0]) == 'A') {
        cout << "Ace";
    }
    else if (toupper(input[0]) == 'J') {
        cout << "Jack";
    }
    else if (toupper(input[0]) == 'Q') {
        cout << "Queen";
    }
    else if (toupper(input[0]) == 'K') {
        cout << "King";
    }
    else if (input[0] == '1') {
        cout << "10";
    }
    else {
        cout << input[0];
    }

    for (int i = 1; i < input.length(); i++) {
        if (toupper(input[i]) == 'H') {
            cout << " of Hearts";
        }
        else if (toupper(input[i]) == 'S') {
            cout << " of Spades";
        }
        else if (toupper(input[i]) == 'D') {
            cout << " of Diamonds";
        }
        else if (toupper(input[i]) == 'C') {
            cout << " of Clubs";
        }
        else {
            cout << input[i];
        }
    }
    return 0;}
```

3. #include <iostream>
using namespace std;

```
int main() {
    float num1, num2, num3;
    cout << "Please enter three numbers: ";
    cin >> num1 >> num2 >> num3;
    cout << "The largest number is: ";
    cout << max({ num1, num2, num3 }) << endl;
    return 0;
}
```

```

4. #include <iostream>
   #include <string>
   using namespace std;

   int main() {
       string grade;
       float grade_num;
       cout << "Enter a letter grade: ";
       cin >> grade;
       if (toupper(grade[0]) == 'A') {
           grade_num = 4;
       }
       else if (toupper(grade[0]) == 'B') {
           grade_num = 3;
       }
       else if (toupper(grade[0]) == 'C') {
           grade_num = 2;
       }
       else if (toupper(grade[0]) == 'D') {
           grade_num = 1;
       }
       else if (toupper(grade[0]) == 'F') {
           grade_num = 0;
       }
       if (grade[1] == '+') {
           if (toupper(grade[0]) == 'A') {
               grade_num = 4;
           }
           else {
               grade_num += 0.3;
           }
       }
       if (grade[1] == '-') {
           grade_num -= 0.3;
       }
       cout << "The numeric value is: " << grade_num << endl;

       return 0;
   }

```

```

5. #include <iostream>
   #include <string>
   using namespace std;

   int main() {
       string grade;
       double grade_num;
       cout << "Enter numerical grade value: ";
       cin >> grade_num;
       for (int i = 0; i < 1; i++) {
           if (grade_num == 4) {
               grade = "A+";
           }
           if (grade_num > 3.7) {
               grade = "A";
           }
       }
   }

```



```

else if (grade_num <= 3.7 && grade_num > 3) {
    grade = "A-";
}
else if (grade_num <= 3 && grade_num > 2.7) {
    grade = "B+";
}
else if (grade_num <= 2.7 && grade_num > 2.3) {
    grade = "B";
}
else if (grade_num <= 2.3 && grade_num > 2) {
    grade = "B-";
}
else if (grade_num <= 2 && grade_num > 1.7) {
    grade = "C+";
}
else if (grade_num <= 1.7 && grade_num > 1.3) {
    grade = "C";
}
else if (grade_num <= 1.3 && grade_num > 1) {
    grade = "C-";
}
else if (grade_num <= 1 && grade_num > 0.7) {
    grade = "D+";
}
else if (grade_num <= 0.7 && grade_num > 0.3) {
    grade = "D";
}
else if (grade_num <= 0.3 && grade_num > 0) {
    grade = "D-";
}
else if (grade_num == 0) {
    grade = "F";
}
else {
    cout << "Invalid grade value entered." << endl;
    i--;
}
}
cout << "Your grade is: " << grade << endl;
return 0;
}

```

Clearly, the code above is ridiculously long and incomprehensible, due to the usage of if-statements, a more practical solution would be to use Switch-Case statements BUT: Switch case statements allow only integer numeric values, so to overcome this, we will cast the numerical grade value into an integer, multiply it by 10 to not lose the 1-decimal point accuracy as well.

Code is shown below 

(NOTE: The code below uses a GCC compiler extension, it might not work on all environments, <https://gcc.gnu.org/onlinedocs/gcc-4.1.2/gcc/Case-Ranges.html> , it works on CodeBlocks though :D)

```

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

int main() {
    string grade;
    double grade_num;
    cout << "Enter numerical grade value: ";
    cin >> grade_num;
    switch (int(grade_num * 10)) {
        case 40:
            grade = "A+";
            break;
        case 37 ... 39:
            grade = "A";
            break;
        case 30 ... 36:
            grade = "A-";
            break;
        case 27 ... 29:
            grade = "B+";
            break;
        case 23 ... 26:
            grade = "B";
            break;
        case 20 ... 22:
            grade = "B-";
            break;
        case 17 ... 19:
            grade = "C+";
            break;
        case 13 ... 16:
            grade = "C";
            break;
        case 10 ... 12:
            grade = "C-";
            break;
        case 7 ... 9:
            grade = "D+";
            break;
        case 3 ... 6:
            grade = "D";
            break;
        case 1 ... 2:
            grade = "D-";
            break;
        case 0:
            grade = "F";
            break;
        default:
            cout << "Invalid grade value entered." << endl;
            return 1; // exit the program with an error code
    }
    cout << "Your grade is: " << grade << endl;
    return 0;
}

```

```

6. #include <iostream>
    #include <string>
    using namespace std;

    int main() {
        string s1, s2, s3;
        cout << "Enter 3 strings: ";
        cin >> s1 >> s2 >> s3;
        cout << "The strings in lexicographical order: ";
        if (s1 < s2 && s1 < s3) {
            cout << s1 << " ";
            if (s2 < s3) {
                cout << s2 << " " << s3 << endl;
            }
            else {
                cout << s3 << " " << s2 << endl;
            }
        }
        else if (s2 < s1 && s2 < s3) {
            cout << s2 << " ";
            if (s1 < s3) {
                cout << s1 << " " << s3 << endl;
            }
            else {
                cout << s3 << " " << s1 << endl;
            }
        }
        else {
            cout << s3 << " ";
            if (s1 < s2) {
                cout << s1 << " " << s2 << endl;
            }
            else {
                cout << s2 << " " << s1 << endl;
            }
        }
    }
}

```

```

7. #include <iostream>
    #include <string>
    using namespace std;

    int main() {
        int year;
        cout << "Enter the year: ";
        cin >> year;
        if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) {
            cout << year << " " << "is a leap year";
        }
        else {
            cout << year << " " << "is not a leap year";
        }
        return 0;
    }
}

```

```

8. #include <iostream>
   #include <string>
   using namespace std;

int main() {
    //REQUIRED CODE:
    int month;
    string monthname;
    int days_number;
    cin >> month;
    if (month == 1 || month == 3 || month == 5 || month == 7 ||
month == 8 || month == 10 || month == 12) {
        days_number = 31;
    }
    else if (month == 4 || month == 6 || month == 9 || month == 11)
{
        days_number = 30;
    }
    else if (month == 2) {
        days_number = 28;
    }
    //BONUS:
    switch (month) {
        case 1: monthname = "January"; break;
        case 2: monthname = "February"; break;
        case 3: monthname = "March"; break;
        case 4: monthname = "April"; break;
        case 5: monthname = "May"; break;
        case 6: monthname = "June"; break;
        case 7: monthname = "July"; break;
        case 8: monthname = "August"; break;
        case 9: monthname = "September"; break;
        case 10: monthname = "October"; break;
        case 11: monthname = "November"; break;
        case 12: monthname = "December"; break;
    }
    cout << "The number of days in " << monthname << " is " <<
days_number << endl;
    return 0;
}

```

```

9. #include <iostream>
   #include <string>
   #include <cmath>
   using namespace std;

int main() {
    double x; // input value
    int n = 0; // count of values
    double sum = 0; // sum of values
    double sum_squared = 0; // sum of squares of values
    while (cin >> x) // read until end of file
    {
        n++; // increment count
        sum += x; // add value to sum
        sum_squared += x * x; // add square of value to sum_squared
    }
    double mean = sum / n; // compute mean
    double stdev = sqrt(sum_squared / n - mean * mean); // compute
standard deviation
    cout << "Count: " << n << endl;
    cout << "Mean: " << mean << endl;
    cout << "Standard deviation: " << stdev << endl;
    return 0;
}

```

NOTE:

- The end of file (referred to on line 11) refers to a condition in the operating system where no more data can be read from the data source. It is not a character but rather a SIGNAL that the OS returns on reaching end of input. To indicate the end of file you can type **CTRL + D** or **CTRL + Z** then hit enter.
- The Standard Deviation formula used in the code is a SIMPLIFIED version of the formula, one that only requires keeping track of the count, the sum, and the sum of squares of the data values as they are read.

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

(ORIGINAL FORMULA)

where σ is the standard deviation, n is the count of values, x_i is the i -th value, and μ is the mean.

$$\sigma = \sqrt{\frac{1}{n} S_{xx} - \mu^2}$$

(ALTERNATIVE FORMULA)

where σ is the standard deviation, n is the count of values, S_{xx} is the sum of squares of values, and μ is the mean.

```

10. #include <iostream>
    using namespace std;

    int main() {
        cout << "This program computes the Fibonacci sequence." << endl;
        int n;
        cout << "Enter the number of terms: ";
        cin >> n;
        if (n == 0 || n == 1) {
            cout << "F(" << n << ") = " << n << endl;
            return 0;
        }
        int f0 = 0, f1 = 1, f2;
        for (int i = 2; i <= n; i++) {
            f2 = f0 + f1;
            f0 = f1;
            f1 = f2;
        }
        cout << "F(" << n << ") = " << f2 << endl;
        return 0;
    }

```

```

11. #include <iostream>
    using namespace std;

    int main() {
        cout << "Enter a number: ";
        int n;
        cin >> n;
        for (int prime = 2; prime <= n; prime++) {
            int count = 0;
            for (int j = 2; j < prime; j++) {
                if (prime % j == 0) {
                    count++;
                    break;
                }
            }
            if (count == 0) {
                cout << prime << " ";
            }
        }
        cout << "are the prime numbers up to " << n << endl;
        return 0;
    }

```

This concludes Sheet (2) Solutions, this document + source code to all programming exercises available on <https://bit.ly/CSE131Sheets>.