SHEET 1 SOLUTIONS

SHEHAB MAHMOUD SALAH I 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 on my **GitHub**: https://bit.ly/CSE131Sheets
> **happy compiling** 😊

(A) Review Questions

1. After properly declaring variables, including <cmath> header, the C++ expressions would possibly look like this;

```cpp
#include <cmath>
s = s0 + v0 * t + 0.5 * g * pow(t, 2);
G = 4 * pow(pi, 2) * pow(a, 3) / (pow(p, 2) * (m1 + m2));
FV = PV * pow((1 + INT) / 100, YRS);
c = sqrt(pow(a, 2) + pow(b, 2) - 2 * a * b * cos(y));
```

2. a) $dm = m * (\sqrt{(1 + v/c)} / \sqrt{(1 - v/c)} - 1)$
   b) $volume = \pi r^2 h$
   c) $volume = (4/3) \pi r^3$
   d) $p = \tan^{-1} (z/\sqrt{(x^2 + y^2)})$

3.

   The quadratic formula is a formula that provides the solution(s) to a quadratic equation of the form `ax^2 + bx + c = 0`, where `a`, `b` and `c` are constants and `a` is not zero. The formula is:

   ```
   x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
   ```

   The problem with your version of the formula is that you are missing parentheses around `2a` in the denominator. This means that you are dividing only by `2` and then multiplying by `a`, which is not correct. You should write:

   ```
   x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{(2a)}
   x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{(2a)}
   ```

4. `n=x;` means that only the integral part of `x` will be assigned to `n`, and the fractional part will be lost. The compiler will give a warning.
   While `n = (int)(x + 0.5);` will guarantee that if the fractional part is less than 0.5, only the integral part will be assigned to `n`, else if 1 will be added to the integral part. The compiler will not give a warning.

5.

Syntax errors:

      1. The semicolon after `int main()`

      2. Forgotten semicolon after `cout` stream.

      3. Wrong `cin` stream construction. Use `>>` instead of `<<`

      4. Forgotten quote in phrase (`The sum of`) and missed (`<<`) after the phrase (`"is: "`)

      5. Return function returns no value.

      6. Properly type `<iostream>`

∴ The correct code would be:

```cpp
#include <iostream>
int main()
{
  int x, y;
  std::cout << "Please enter two numbers:";
  std::cin >> x >> y;
  std::cout << "The sum of " << x << " and " << y
          << " is: " << x + y << "\n";
  return 0;
}
```

6. Logic errors:

      1. Non-initialized variable total.
      2. Line 9, assigned (total+x1) instead of (total+x2)
      3. Line 10, total / 2 will produce integral value, to get the whole result total: float average = total / 2.0

∴ The correct code would be:

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

int main()
{
        int total=0;
        int x1;
                cout << "Please enter a number: ";
                cin >> x1;
        total = total + x1;
                cout << "Please enter another number: ";
        int x2;
                cin >> x2;
        total = total + x2;
        float average = total / 2;
                cout << "The average of the two numbers is "<< average << "\n";
        return 0;

}
```

7. a. will output "4"
   b. will output "22"

8. a. fname: James , lname: Carter , age: 56
   b. fname: Lyndon , lname: Johnson , age: 49
   c. fname: Hodding , lname: Carter , age: 0
   d. fname: Richard , lname: M, age: 0

9. a. 6.25
   b. 6
   c. 12.5
   d. 1.414
   e. 3
   f. "HelloWorld"
   g. "WorldHello"
   h. -3
   i. "el"
   j. 10

(B) Programming Exercises

1.

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

int main()
{
    int x, y;
    cout << "Please enter two integers\n";
    cin >> x >> y;
    cout << "The sum is " << x + y << "\n";
    cout << "The difference is " << x - y << "\n";
    cout << "The product is " << x * y << "\n";
    cout << "The average is " << float(x + y) / 2.0 << "\n";
    cout << "The distance is " << abs(x - y) << "\n";
    cout << "The maximum is " << max(x,y) << "\n";
    cout << "The minimum is " << min(x,y) << "\n";

    return 0;

}
```

2.

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

int main()
{
        double pi = 3.14159265358979323846;
        double r;
        cout << "Please enter the radius\n";
        cin >> r; cout << "\n";
        cout << "The area of the circle = " << pi * r * r << "\n";
        cout << "The circumference of the circle = " << 2 * pi * r << "\n";
        cout << "The volume of the sphere = " << (1.333333) * pi * pow(r, 3) << "\n";
        cout << "The surface area of the sphere = " << (4) * pi * pow(r, 2) << "\n";


        return 0;

}
```

3.

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

int main()
{
        double l, w;
        cout << "Please enter the rectangle's length\n";
        cin >> l; cout << "\n";
        cout << "Please enter the rectangle's width\n";
        cin >> w; cout << "\n";
        cout << "The rectangle's area = " << l * w << "\n";
        cout << "The rectangle's perimeter = " << 2*(l + w) << "\n";
        cout << "The length of the rectangle's diagonal = " << sqrt(pow(l, 2) + pow(w, 2))
<< "\n";
        //The pythagorean theorem states that d^2=l^2+w^2 where "d" is the diagonal's
length

        return 0;

}
```

4.

```cpp
#include <iostream>
#include <cmath>

using namespace std;
//function to convert a given decimal angle to radian:
    double convert(double degree) {
        double pi = 3.14159265359;
        return (degree * (pi / 180));
}
int main()
{
    double a, degree1, degree2{};
    cout << "Please enter the length of one of the traingle's sides\n";
    cin >> a;
    cout << "Please enter the two adjacent angles (in degrees) to this side\n";
    cin >> degree1 >> degree2; cout << "\n";
    double radian1, radian2, radian3;
    radian1 = convert(degree1); radian2 = convert(degree2); radian3 = 3.14159265359 -
(radian1 + radian2);
    cout << "The lengths of the other two sides = " << (a * sin(radian1) /
sin(radian3)) << " , " << (a * sin(radian2) / sin(radian3)) << "\n\n";
    cout << "The size of the third angle = " << radian3 * 57.2958 << " degrees"
<<"\n";

    return 0;

}
```

*Code written in IDE environment for easier reading and following:

```cpp
#include <iostream>
#include <cmath>

using namespace std;
//function to convert a given decimal angle to radian:
    double convert(double degree) {
        double pi = 3.14159265359;
        return (degree * (pi / 180));
}
int main()
{
    double a, degree1, degree2{};
    cout << "Please enter the length of one of the traingle's sides\n";
    cin >> a;
    cout << "Please enter the two adjacent angles (in degrees) to this side\n";
    cin >> degree1 >> degree2; cout << "\n";
    double radian1, radian2, radian3;
    radian1 = convert(degree1); radian2 = convert(degree2); radian3 = 3.14159265359 - (radian1 + radian2);
    cout << "The lengths of the other two sides = " << (a * sin(radian1) / sin(radian3)) << " , " << (a * sin(radian2) / sin(radian3)) << "\n\n";
    cout << "The size of the third angle = " << radian3 * 57.2958 << " degrees" <<"\n";

    return 0;

}
```

*Debug Console:

```
Microsoft Visual Studio Debug Console
Please enter the length of one of the traingle's sides
15
Please enter the two adjacent angles (in degrees) to this side
24
59

The lengths of the other two sides = 6.14687 , 12.9541

The size of the third angle = 97 degrees
```

5.

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

int main() {
        int n;
        cout << "Please enter an integer >= 1000: \n";
        cin >> n;
        cout << n / 1000 << "," << n % 1000;

        return 0;
}
```

6.

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

int main() {
        int hour, minute;
        int due_hour, due_minute;
        int current_time, due_time;
        int diff_time;

        //ask for due date of the next assignment
        cout << "What is the due date of the next assignment?\n";
        cout << "Hour of due date: ";
        cin >> due_hour;
        cout << "Minute of due date: ";
        cin >> due_minute; cout << "\n";

        //convert due date to minutes
        due_time = due_hour * 60 + due_minute;

        //get current time from system
        time_t now = time(0);
        tm* ltm = localtime(&now);
        hour = ltm->tm_hour;
        minute = ltm->tm_min;

        //convert current time to minutes
        current_time = hour * 60 + minute;

        //calculate difference between current time and due time
        diff_time = due_time - current_time;

        //print result
        cout << "The number of minutes between the current time and the due date is " <<
diff_time << "\n";

        return 0;
}
```

BE CAREFUL COMPILING THIS CODE ON YOUR MACHINE, THE FUNCTION `localtime` IS UNSAFE, IT IS PREFERABLE TO USE AN ONLINE COMPILER OR A VIRTUAL MACHINE.