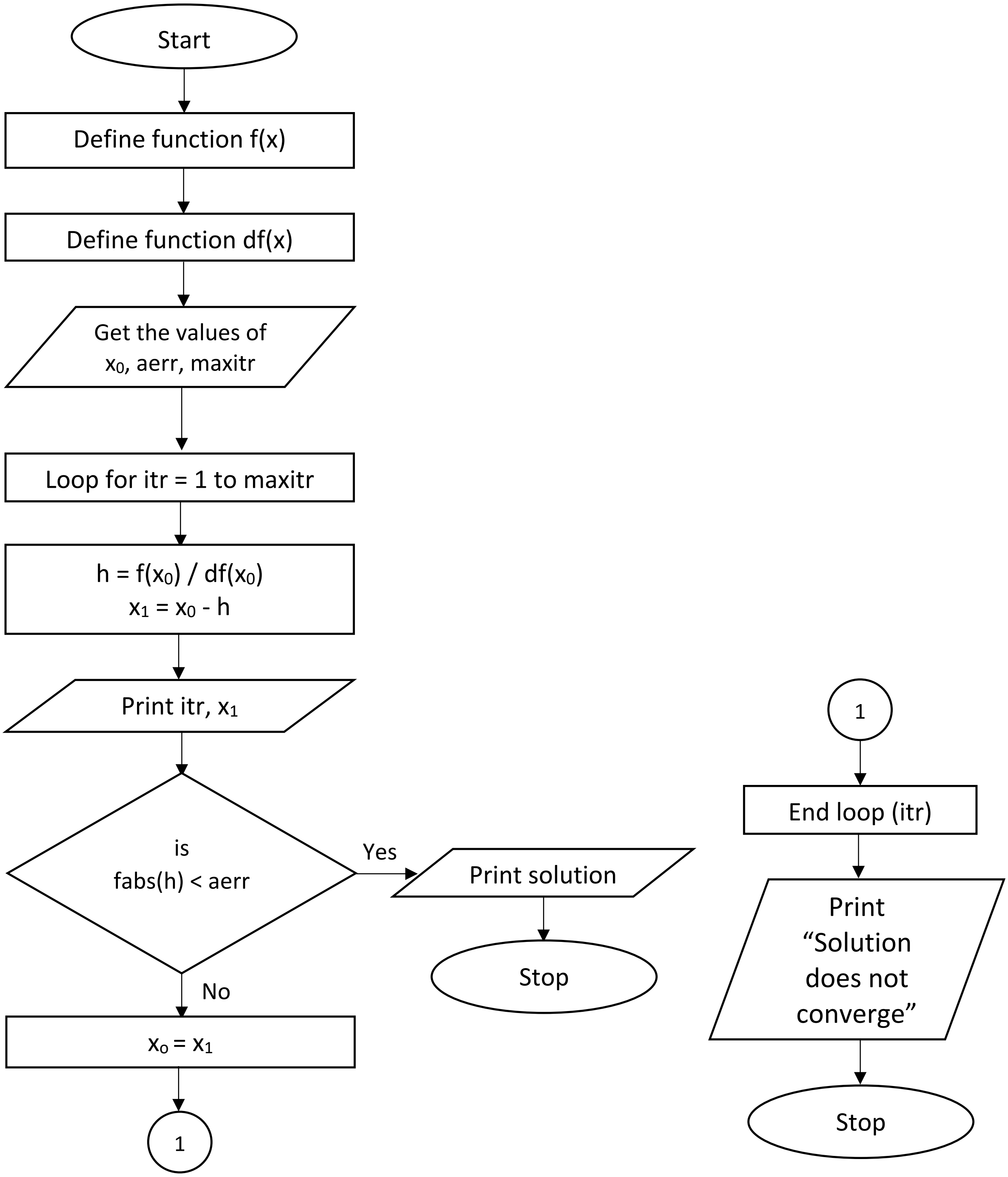
Date -
26/09/2022

PRACTICAL 2

- **Objective:** To write a C++ program to find the root of the equation (**$x\log_{10}x - 1.2$**) using Newton Raphson method.
- **Flow Chart:**



- **Practical Code:**

```
#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

float f(float x) {
   return x*log10(x) - 1.2;
}

float df(float x) {
   return log10(x) + 0.43429;
}

int main() {
   int itr, maxitr;
   float h, x0, x1, aerr;
   cout << "Enter x0, allowed error, maximum iterations : " << endl;
   cin >> x0 >> aerr >> maxitr;
   cout << fixed;
   for (itr=1; itr<=maxitr; itr++) {
      h = f(x0)/df(x0);
      x1 = x0 - h;
      cout << "Iteration no. " << itr << ", x = " << x1 << endl;
      if (fabs(h) <  aerr) {
         cout << "After no. " << itr << "iterations, root = " << x1;
         return 0;
      }
      x0 = x1;
   }
   cout << "Iterations not sufficient, solution does not converge" << endl;

   return 1;
}
```

- **Output:**

```
Enter x0, allowed error, maximum iterations :
2 0.000001 10
Iteration no. 1, x = 2.813170
Iteration no. 2, x = 2.741109
Iteration no. 3, x = 2.740646
Iteration no. 4, x = 2.740646

After 4 iterations, root = 2.740646
```

- **Application:**

a. Solving nonlinear equations such as Kepler’s equation a + bsin x = x for constants a and b in celestial mechanics.

b. Computing 1/√x for video games. (This is needed to rescale vectors to have length 1.)

c. Solve equations that occur in GPS calculations.

d. Inverse kinematic problems (robotics, video games animation).