**Drone Simulator Software**

*SARAH SHAHID.*

This is not the final code, just my version of it that shows what logic I followed.  
Most of the functions that were required have been written.

**Rough Draft of CODE:**

//prototyping, above main.

//calling, inside main

//definition, below main.

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

//prototyping

void startDay();

void deliverpackage(string, int&, int&, int&, int&); // for all three A, B and C.

int checkweather();

string tellweather(int);

bool checkobstacle();

string tellobstacle(bool);

int batterydrain();

void tellbatterydrain(int);

void displaySummary(int&, int&, int&, int&);

int main() {

//initializing variables.

int battery = 100;

int successful = 0;

int delayed = 0;

int failed = 0;

//calling functions

srand(time(0));

startDay();

deliverpackage("A", battery, successful, delayed, failed);

deliverpackage("B", battery, successful, delayed, failed);

deliverpackage("C", battery, successful, delayed, failed);

displaySummary(battery, successful, delayed, failed);

return 0;

}

void startDay()

{

cout << "GOOD MORNING.\n";

cout << "Battery = 100%\n";

cout << "Your deliveries for today are as follows:\n1. Location A\n2. Location B\n3. Location C\n\n";

}

//call all variables by reference so that their original values can be modified.

void deliverpackage(string location, int &battery, int &successful, int &delayed, int &failed)

{

cout << "Checking conditions for location " << location << ". " << endl;

int x = checkweather();

cout << tellweather(x);

bool obs = checkobstacle();

cout << tellobstacle(obs);

int level = batterydrain();

tellbatterydrain(level);

cout << endl;

//1 is sunny, 2 is rainy, 3 is windy.

if (x == 2)

{

cout << "Delivery delayed.\n";

cout << "Remaining battery = " << battery << "%" << endl;

cout << "-----------------------------------------------\n";

delayed += 1;

}

else if (x == 3 && battery < 40)

{

cout << "The drone is returning for a recharge.\n";

cout << "The delivery has failed.\n";

battery += 10;

cout << "Remaining battery = " << battery << "%" << endl;

cout << "-----------------------------------------------\n";

failed += 1;

}

else if (obs)

{

cout << "The flight has been rerouted. (Extra battery drainage)\n";

cout << "The delivery is successful.\n";

battery -= 5; //extra battery drainage due to reroute.

battery -= level;

cout << "Remaining battery = " << battery << "%" << endl;

cout << "-----------------------------------------------\n";

successful += 1;

}

else

{

cout << "The delivery is successful.\n";

battery -= level;

successful += 1;

cout << "Remaining battery = " << battery << endl;

cout << "-----------------------------------------------\n";

}

}

int checkweather()

{

//returns a random value between 1 and 3.

return rand() % 3 + 1;

}

string tellweather(int i)

{

if (i == 1)

{

return "Weather Status: Sunny\n";

}

else if (i == 2)

{

return "Weather Status: Rainy\n";

}

else if (i == 3)

{

return "Weather Status: Windy\n";

}

}

bool checkobstacle()

{

//returns either 0 or 1.

return rand() % 2;

}

string tellobstacle(bool i)

{

if (i) //when i is 1 (true)

{

return "Obstacle Status: DETECTED.\n";

}

else

{

return "Obstacle Status: NOT DETECTED.\n";

}

}

int batterydrain()

{

//generates a random value between 10 and 30.

return rand() % (30 - 10 + 1) + 10;

}

void tellbatterydrain(int i)

{

cout << "The battery drain for this flight would be " << i << "%." << endl;

}

//function to display all results at the end.

void displaySummary(int& battery, int& successful, int& delayed, int& failed)

{

cout << "FLIGHT SUMMARY:\n";

cout << "Battery remaining: " << battery << "%\n";

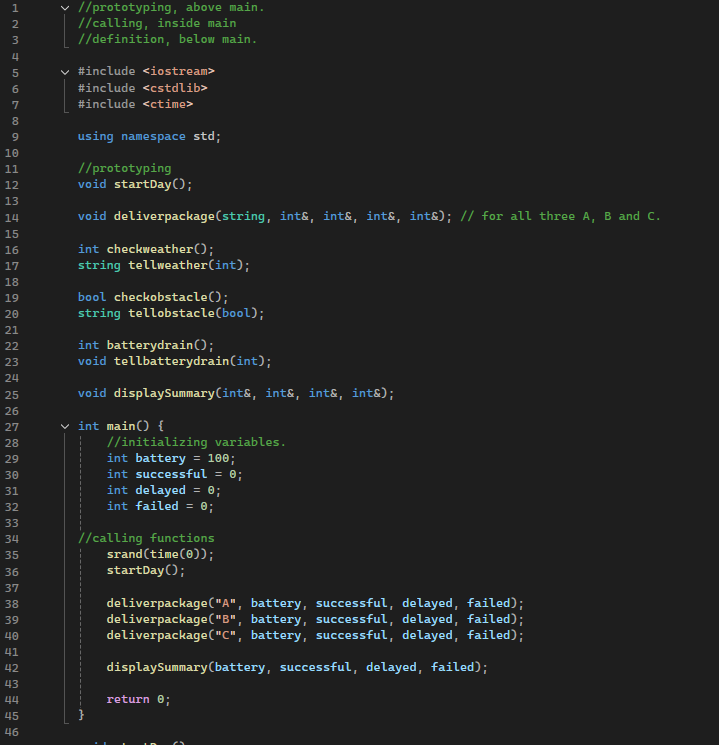
cout << "Total deliveries = 3\n";

cout << "Deliveries successful = " << successful << endl;

cout << "Deliveries delayed = " << delayed << endl;

cout << "Deliveries failed = " << failed << endl;

}

****

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of a program

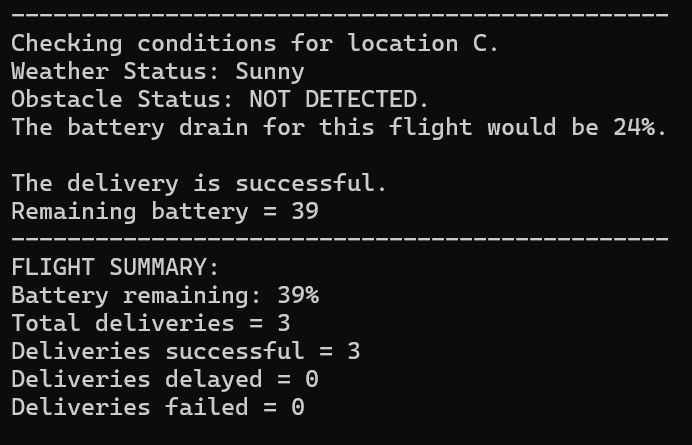
AI-generated content may be incorrect.**

**CODE OUTPUT:**

1st Sample Run:

A screenshot of a black screen

AI-generated content may be incorrect.



2nd Sample Run:

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

3rd Sample Run:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.