**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <ctype.h>

#define MIN\_RAND\_MINUTES\_FACTOR 1.2

#define MAX\_RAND\_MINUTES\_FACTOR 1.5

#define SENTINEL\_VALUE -1

double getValidDouble(int min, int max, int sentinel);

double calculateFare(double base, double minuteCost, double mileCost, double minRate, double miles, int minutes);

void printFare(int count, double miles, int minutes, double fare);

int main(void) {

// Initialize variables

int minMiles = 1;

int maxMiles = 100;

double baseFare = 1.8;

double costPerMinute = 0.25;

double costPerMile = 1.2;

double minFlatRate = 20.0;

// Seed for random number generation

srand(time(NULL));

// Initialize ride counter and totals

int riderNumber = 1;

int totalRides = 0;

double totalMiles = 0, totalMinutes = 0, totalFare = 0;

while (1) {

printf("\nWelcome to the UCCS Ride Share. We can only provide services for rides from %d to %d miles.", minMiles, maxMiles);

double miles = getValidDouble(minMiles, maxMiles, SENTINEL\_VALUE);

if (miles == SENTINEL\_VALUE && totalRides == 0) {

printf("\nUCCS Ride Share Business Summary\n");

printf("\nRider Number of Miles Number of Minutes Ride Fare Amount\n");

printf("-------------------------------------------------------------\n");

printf("\n\t\tThere were no rides\n");

break;

}

else if (miles == SENTINEL\_VALUE && totalRides != 0) {

printf("\nUCCS Ride Share Business Summary\n");

printf("Rider Number of Miles Number of Minutes Ride Fare Amount\n");

printf("-------------------------------------------------------------\n");

printFare(riderNumber, totalMiles, totalMinutes, totalFare);

break;

}

int minMinutes = (int)(MIN\_RAND\_MINUTES\_FACTOR \* miles);

int maxMinutes = (int)(MAX\_RAND\_MINUTES\_FACTOR \* miles);

int minutes = minMinutes + rand() % (maxMinutes - minMinutes + 1);

double fare = calculateFare(baseFare, costPerMinute, costPerMile, minFlatRate, miles, minutes);

printf("\nCurrent Ride Information\n");

printf("Rider Number of Miles Number of Minutes Ride Fare Amount\n");

printFare(riderNumber, miles, minutes, fare);

printf("\n\tThanks for riding with us\n");

// Update totals

totalRides++;

totalMiles += miles;

totalMinutes += minutes;

totalFare += fare;

riderNumber++;

}

return 0;

}

double getValidDouble(int min, int max, int sentinel) {

double value;

char suffix;

int c;

while(1) {

printf("\nEnter the number of miles to your destination: ");

if (scanf("%lf%c", &value, &suffix) != 2 || (suffix != '\n' && suffix != 'm') || value < min || value > max) {

if (value == sentinel) {

break;

}

else if (value < min || value > max) {

printf("Error: ");

printf("Not within %d and %d miles. ", min, max);

} else {

printf("Error: ");

printf("You didn't enter the number of miles correctly. ");

while ((c = getchar()) != '\n' && c != EOF);

}

while ((c = getchar()) != '\n' && c != EOF);

} else {

break;

}

}

return value;

}

double calculateFare(double base, double minuteCost, double mileCost, double minRate, double miles, int minutes) {

double rideFare = base + (minuteCost \* minutes) + (mileCost \* miles);

return (rideFare < minRate) ? minRate : rideFare;

}

void printFare(int count, double miles, int minutes, double fare) {

printf("%-6d %-18.1f %-20d $%-17.2f\n", count, miles, minutes, fare);

}

**Output**:

A screenshot of a computer screen

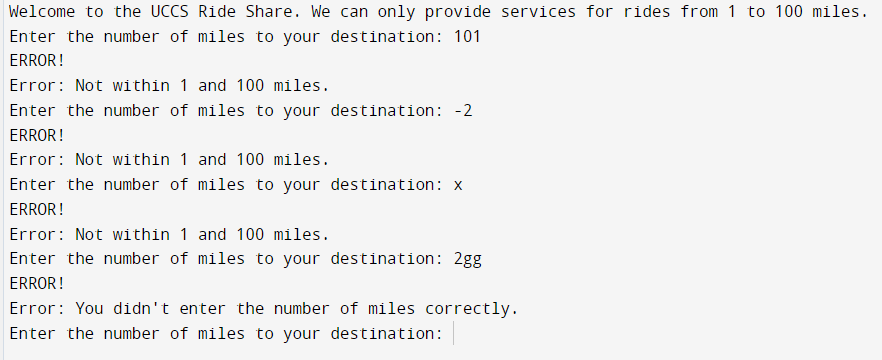
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated



A screen shot of a computer

Description automatically generated