**WEEK 8:**

**CODE:**

#include <iostream>

#include <iomanip>

#include <vector>

#include <string>

struct Boat {

int boatNumber;

double moneyTaken;

double totalHoursHired;

std::string returnTime;

};

const double HOURLY\_RATE = 20.0;

const double HALF\_HOUR\_RATE = 12.0;

const int NUM\_BOATS = 10;

const std::string OPEN\_TIME = "10:00";

const std::string CLOSE\_TIME = "17:00";

void calculateMoneyForBoat(Boat& boat) {

std::cout << "Enter the hours you want to hire Boat " << boat.boatNumber << " (between 0.5 and 7): ";

double hoursToHire;

std::cin >> hoursToHire;

while (hoursToHire < 0.5 || hoursToHire > 7 || boat.returnTime.compare(CLOSE\_TIME) >= 0) {

std::cout << "Invalid hours or boat cannot be hired after " << CLOSE\_TIME << ". Please enter a valid duration: ";

std::cin >> hoursToHire;

}

double cost;

if (hoursToHire <= 1) {

cost = HOURLY\_RATE \* hoursToHire;

} else {

cost = HALF\_HOUR\_RATE \* hoursToHire;

}

boat.moneyTaken += cost;

boat.totalHoursHired += hoursToHire;

size\_t colonPos = boat.returnTime.find(':');

int currentHour = std::stoi(boat.returnTime.substr(0, colonPos));

int currentMinute = std::stoi(boat.returnTime.substr(colonPos + 1));

currentHour += static\_cast<int>(hoursToHire);

boat.returnTime = std::to\_string(currentHour) + ":" + std::to\_string(currentMinute);

std::cout << "Boat " << boat.boatNumber << " hired for " << hoursToHire << " hours. Total cost: $" << cost << "\n";

}

void findAvailableBoats(const std::vector<Boat>& boats, const std::string& currentTime) {

std::cout << "\n--- Available Boats and Slots ---\n";

for (const auto& boat : boats) {

if (boat.returnTime.compare(currentTime) <= 0) {

std::cout << "Boat " << boat.boatNumber << " is available until " << boat.returnTime << "\n";

}

}

}

void calculateTotalMoney(const std::vector<Boat>& boats) {

double totalMoney = 0.0;

double totalHours = 0.0;

int unusedBoats = 0;

int mostUsedBoat = 1;

double maxHoursHired = boats[0].totalHoursHired;

for (const auto& boat : boats) {

totalMoney += boat.moneyTaken;

totalHours += boat.totalHoursHired;

if (boat.totalHoursHired == 0) {

++unusedBoats;

}

if (boat.totalHoursHired > maxHoursHired) {

mostUsedBoat = boat.boatNumber;

maxHoursHired = boat.totalHoursHired;

}

}

std::cout << "\n--- End of Day Report ---\n";

std::cout << "Total revenue from all boats: $" << totalMoney << "\n";

std::cout << "Total hours boats were hired: " << totalHours << " hours\n";

std::cout << "Number of boats not used today: " << unusedBoats << "\n";

std::cout << "Boat #" << mostUsedBoat << " was used the most, with " << maxHoursHired << " hours hired.\n";

}

int main() {

std::vector<Boat> boats;

for (int i = 1; i <= NUM\_BOATS; ++i) {

boats.push\_back({i, 0.0, 0.0, OPEN\_TIME});

}

std::cout << "Enter the current time (24-hour format, e.g., 14:30): ";

std::string currentTime;

std::cin >> currentTime;

while (true) {

std::cout << "\nOptions:\n";

std::cout << "1. Hire a boat\n";

std::cout << "2. View available boats and slots\n";

std::cout << "3. View end-of-day report\n";

std::cout << "4. End the program\n";

std::cout << "Enter your choice (1-4): ";

int choice;

std::cin >> choice;

if (choice == 1) {

int boatNumber;

std::cout << "Enter the boat number you want to hire: ";

std::cin >> boatNumber;

if (boatNumber >= 1 && boatNumber <= NUM\_BOATS) {

calculateMoneyForBoat(boats[boatNumber - 1]);

} else {

std::cout << "Invalid boat number. Please enter a number between 1 and " << NUM\_BOATS << ".\n";

}

} else if (choice == 2) {

findAvailableBoats(boats, currentTime);

} else if (choice == 3) {

calculateTotalMoney(boats);

break; // End the program after generating the report

} else if (choice == 4) {

break; // End the program

} else {

std::cout << "Invalid choice. Please enter a number between 1 and 4.\n";

}

}

return 0;

}

**OUTPUT:**



