Worksheet 1

Task 1

1.

#include <iostream>

using namespace std;

class Temp

{

float input,output;

int choice;

public:

void getInput()

{

cout<<"Enter the temperature you want to convert: ";

while(!(cin>> input))

{

cout << "Your input is INVALID!!! Enter the temperature: ";

cin.clear(); // Clear error flags

cin.ignore(123, '\n'); // Discard invalid input

}

cout<<endl;

displayoptions();

}

void displayoptions()

{

cout<<"1. Fahrenheit to Celsius"<<endl;

cout<<"2. Celsius to Fahrenheit"<<endl;

cout<<"3. Press any key to exit"<<endl;

getChoice();

}

void getChoice()

{

cout<<endl;

cout<<"Enter the temperature for conversion: ";

cin>> choice;

calculateConversion();

}

void calculateConversion()

{

if(choice ==2)

{

getCelsius();

}

if(choice ==1)

{

getFahrenheit();

}

}

void getCelsius()

{

output = (input-32)/1.8;

displayData();

}

void getFahrenheit()

{

output = 1.8\*input+32;

displayData();

}

void displayData()

{

cout<<endl;

cout<<"Here you have converted temperature : "<< output;

cout<<endl<<endl;

getInput();

}

};

int main()

{

Temp t1;

t1.getInput();

}

2.

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int generateRandomNumber(int min, int max) {

return rand() % (max - min + 1) + min;

}

int main() {

srand(time(nullptr));

int choice;

int min, max;

cout << "..........Number Guessing Game.........." << endl<<endl;

cout << " ->Select Difficulty Level:" << endl;

cout << "1. Easy (1-8)" << endl;

cout << "2. Medium (1-30)" << endl;

cout << "3. Hard (1-50)" << endl<<endl;

cout << "Enter the level you want to play: ";

cin >> choice;

switch(choice) {

case 1:

min = 1;

max = 8;

break;

case 2:

min = 1;

max = 30;

break;

case 3:

min = 1;

max = 50;

break;

default:

cout << "Invalid choice!!! Getting out!!!" << endl;

return 1;

}

int secretNumber = generateRandomNumber(min, max);

int guess;

int attempts = 0;

cout << "Guess the SECRET number between " << min << " and " << max << endl;

do {

cout << "Enter your GUESS: ";

cin >> guess;

attempts++;

if (guess < secretNumber) {

cout << "Too down! Go up." << endl;

} else if (guess > secretNumber) {

cout << "Too up! Go down." << endl;

} else {

cout << "Congratulations! You've guessed the SECRET number (" << secretNumber << ") in " << attempts << " attempts." << endl;

break;

}

} while (true);

return 0;

}

3.

#include <iostream>

using namespace std;

int main() {

int numbers[100];

int numElements;

cout << "Enter the number of elements: ";

cin >> numElements;

cout << "Enter " << numElements << " integer numbers:\n";

for (int i = 0; i < numElements; ++i) {

cin >> numbers[i];

}

int min = numbers[0];

int max = numbers[0];

for (int i = 1; i < numElements; ++i) {

if (numbers[i] < min) {

min = numbers[i];

}

if (numbers[i] > max) {

max = numbers[i];

}

}

cout << "Minimum value: " << min << endl;

cout << "Maximum value: " << max << endl;

return 0;

}

4.

#include <iostream>

#include <string>

using namespace std;

int main()

{

string password;

bool has\_upper = false, has\_lower = false, has\_digit = false, has\_punct = false;

cout << "Enter your password: ";

getline(cin, password);

for (char c : password)

{

if (isupper(c))

{

has\_upper = true;

}

if (islower(c))

{

has\_lower = true;

}

if (isdigit(c))

{

has\_digit = true;

}

if (ispunct(c))

{

has\_punct = true;

}

}

if (has\_upper && has\_lower && has\_digit && has\_punct && password.length() >= 8)

{

cout << "Congrats!! Your Password is strong." << endl;

}

else

{

cout << "Your Password is weak." << endl;

}

return 0;

}

Task 2

#include <iostream>

using namespace std;

class ClassAtt {

int noOfStd=0;

int choice;

int count;

int indexOfStd;

char status;

char atd[];

public:

void showMenu()

{

cout<<endl<<endl<<endl<<endl;

cout<<" Student Attendance: "<<endl<<endl;

cout<<"Press 1 to add Students."<<endl;

cout<<"Press 2 to mark Student's attendance."<<endl;

cout<<"Press 3 to display Student's overall attendance percentage."<<endl;

cout<<"Press 4 to exit."<<endl<<endl;

getChoice();

}

void getChoice()

{

cout<<"Enter your choice: ";

cin>>choice;

cout<<endl;

decideOpt();

}

void decideOpt()

{

if(choice==1)

{

cout << "Enter the number of students you want to add: ";

cin >> count;

addStd(count);

}

else if(choice ==2)

{

cout<<"Out of "<<noOfStd<<" students, Enter the student number you want to mark: ";

cin>> indexOfStd;

cout<<"Press 'P' for Present"<<endl;

cout<<"Press 'A' for Absent"<<endl;

cout<<"Enter: ";

cin>>status;

markAtd();

}

else if(choice ==3)

{

overallAtdPer();

}

else

{

cout<<"Exited";

}

}

void addStd(int count)

{

if(count>0)

{

noOfStd +=count;

cout<< count <<" number of students added successfully."<<endl<<endl;

}

else{

cout<<"Invalid number. Enter a positive number."<<endl<<endl;

}

atd[noOfStd];

showMenu();

}

void markAtd()

{

if(indexOfStd>=1 && indexOfStd<=noOfStd)

{

if(status =='p'|| status =='P'|| status =='a'|| status =='A'){

atd[indexOfStd -1] = status;

cout<<"Attendance marked successfully.";

}

else

{

cout<<"Invalid attendance status. Enter correct status next time.";

}

}

else

{

cout<<"Invalid student number. Enter correct student number next time.";

}

showMenu();

}

void overallAtdPer() {

float sumOfPresent = 0;

float sumOfAbsent = 0;

float sumOfNotMarked = 0;

for (int i = 0; i < noOfStd; i++) {

if (atd[i] == 'p' || atd[i] == 'P') {

sumOfPresent++;

} else if (atd[i] == 'a' || atd[i] == 'A') {

sumOfAbsent++;

} else {

sumOfNotMarked++;

}

}

float presentPer = (sumOfPresent / noOfStd) \* 100.0;

float absentPer = (sumOfAbsent / noOfStd) \* 100.0;

float notMarkedPer = (sumOfNotMarked / noOfStd) \* 100.0;

cout << endl << endl;

cout << "Out of " << noOfStd << " Students " << presentPer << "%" << " students are present." << endl;

cout << "Out of " << noOfStd << " Students " << absentPer << "%" << " students are absent." << endl;

cout << "Out of " << noOfStd << " Students " << notMarkedPer << "%" << " students's attendance is not marked." << endl;

showMenu();

}

};

int main()

{

ClassAtt c1;

c1.showMenu();

return 0;

}

Task 3

#include <iostream>

#include <string>

using namespace std;

// This is Structure to represent item.

struct tbcItem

{

int productID;

string name;

int quantity;

float price;

};

//This is Array to store inventory items, with a maximum capacity.

const int MAX\_ITEMS = 100;

tbcItem inventory[MAX\_ITEMS];

int itemCount = 0; // This tracks the current number of items in the inventory.

// This is the Function to add item to the inventory.

void addItem()

{

if (itemCount >= MAX\_ITEMS)

{

cout << "Inventory is full!!! Cannot add more items now." << endl;

return;

}

tbcItem newItem;

cout << "Enter Product ID: ";

cin >> newItem.productID;

cout << "Enter Product Name: ";

cin.ignore(); // To consume the '\n' character left in the buffer

getline(cin, newItem.name);

cout << "Enter Quantity of the product: ";

cin >> newItem.quantity;

cout << "Enter Price of the product: ";

cin >> newItem.price;

// This Checks for unique Product ID.

for (int i = 0; i < itemCount; ++i)

{

if (inventory[i].productID == newItem.productID)

{

cout << " Item with this Product ID already exists in this inventory." << endl;

return;

}

}

inventory[itemCount++] = newItem;

cout << "Item is added successfully." << endl;

}

// This is the Function to display all items in the inventory.

void displayItems()

{

if (itemCount == 0)

{

cout << "This Inventory is empty." << endl;

return;

}

for (int i = 0; i < itemCount; i++) {

cout << "Product ID: " << inventory[i].productID

<< ", Name: " << inventory[i].name

<< ", Quantity: " << inventory[i].quantity

<< ", Price: Rs." << inventory[i].price << endl;

}

}

// This is the Function to find an item's index by its Product ID.

int findItemIndex(int productID)

{

for (int i = 0; i < itemCount; i++)

{

if (inventory[i].productID == productID)

{

return i;

}

}

return -1; // This Returns -1 if the item is not found in the inventory.

}

// This is the Function to remove an item from the inventory.

void removeItem(int productID)

{

int index = findItemIndex(productID);

if (index == -1)

{

cout << "Item with this Product ID " << productID << " cannnot be found." << endl;

return;

}

for (int i = index; i < itemCount - 1; i++)

{

inventory[i] = inventory[i + 1];

}

itemCount--;

cout << "Item removed successfully from the inventory." << endl;

}

// This is the Function to update an item's quantity and price in the inventory.

void updateItem(int productID)

{

int index = findItemIndex(productID);

if (index == -1)

{

cout << "Item with this Product ID " << productID << " cannot be found." << endl;

return;

}

cout << "Enter new Quantity: ";

cin >> inventory[index].quantity;

cout << "Enter the new Price: ";

cin >> inventory[index].price;

cout << "Item updated successfully!!!!!!!" << endl;

}

int main()

{

int choice, productID;

do

{

cout << "\n.....Inventory System.....\n"

<< "1. Add Item\n"

<< "2. Remove Item\n"

<< "3. Update Item\n"

<< "4. Display Items\n"

<< "5. Exit\n"

<< "Enter your choice: ";

cin >> choice;

switch (choice)

{

case 1:

addItem();

break;

case 2:

cout << "Enter Product ID to remove: ";

cin >> productID;

removeItem(productID);

break;

case 3:

cout << "Enter Product ID to update: ";

cin >> productID;

updateItem(productID);

break;

case 4:

displayItems();

break;

case 5:

cout << "This already exists in the Inventory System. Enter new!!" << endl;

break;

default:

cout << "Invalid choice!! Please try again." << endl;

}

} while (choice != 5);

return 0;

}