**Q1:**

#include <iostream>

#include <vector>

#define *STUDENT\_ID* "23K0018"

#define *DEFAULT\_LEVEL* "Low"

#define *DEFAULT\_COST* 300

#define *MAX\_PORTS* 23

using namespace std;

const *string* protocols\_list[6] = { "HTTPS", "FTP", "UDP", "ICMP", "SSH", "SNMP" };

class *SecurityTool* {

protected:

*string* securityLevel;

    int numDevices;

    float cost;

public:

*SecurityTool*(*string* *level*, int *devices*, float *cost*)

        : *numDevices*(*devices*), *cost*(*cost* <= 0 ? *DEFAULT\_COST* : *cost*)

    {

        if (!(*level* *==* "High" || *level* *==* "Medium" || *level* *==* "Low"))

            this->securityLevel *=* *DEFAULT\_LEVEL*;

        else

            this->securityLevel *=* *level*;

    }

    void *performScan*() {

        cout *<<* "generic-security scan performed" *<<* *endl*;

    }

};

class *FirewallTool* : public *SecurityTool* {

private:

*vector*<int> ports;

*vector*<*string*> protocols;

public:

*FirewallTool*(*string* *level*, float *cost*) : *SecurityTool*(*level*, 10, *cost*) { this->*generateList*(); }

    void *generateList*() {

        int digit = *STUDENT\_ID*[6]-'0';

        ports.*reserve*(23);

        for (int i = 1; i <= *MAX\_PORTS*; i++)

            ports.*push\_back*(digit + i);

    }

    void *addProtocol*(*string* *protocol*) {

        for (int i = 0; i < 6; i++) {

            if (*protocol* *==* protocols\_list[i]) {

                protocols.*push\_back*(*protocol*);

                return;

            }

        }

        cout *<<* "Invalid protocol" *<<* *endl*;

    }

    void *performScan*() {

        cout *<<* "Performing scan level [" *<<* securityLevel *<<* "]:\n";

        cout *<<* "Allowing traffic from [protocols]:\n";

        for (*string* protocol : protocols)

            cout *<<* protocol *<<* '\n';

        if (securityLevel *==* "Low")

            cout *<<* "TCP\nDNS\n";

        cout *<<* "Allowing traffic from [ports]:\n";

        for (int port : ports)

            cout *<<* port *<<* '\n';

        if (securityLevel *!=* "High") {

            int last = ports*[*22*]*;

            for (int i = 1; i <= (securityLevel *==* "Medium" ? 2 : 5); i++)

                cout *<<* last+i *<<* '\n';

        }

    }

};

int *main*() {

    cout *<<* "Fasih Hasan Khan\n23k0018\n\n";

*FirewallTool* *tool*("Medium", 10);

    tool.*addProtocol*("HTTPS");

    tool.*performScan*();

    return 0;

}

**OUTPUT:**

**A screenshot of a computer

Description automatically generated**

**Q2:**

#include <iostream>

#include <vector>

using namespace std;

const *string* default\_weapons[5] = { "Sword", "Axe", "Bow", "Spear", "Hammer" };

const *string* experience\_str[3] = { "Beginner", "Intermediate", "Advanced" };

class *Player* {

private:

    int id, health;

*string* name;

public:

*Player*(int *id*, *string* *name*) : *id*(*id*), *name*(*name*), *health*(100) {}

    void *setHealth*(int *health*) { this->health = *health*; }

    int *getHealth*() { return this->health; }

};

class *Weapon* {

private:

*vector*<*string*> weapons;

public:

*Weapon*(*vector*<*string*> *weapons*) : *weapons*(*weapons*.*begin*(), *weapons*.*end*()) {

        if (*weapons*.*size*() < 5) {

            for (int i = 0; i < 5-*weapons*.*size*(); i++)

                this->weapons.*push\_back*(default\_weapons[i]);

        }

    }

    void *display*() {

        for (int i = 0; i < this->weapons.*size*(); i++)

            cout *<<* i+1 *<<* ": " *<<* this->weapons*[*i*]* *<<* '\n';

        cout *<<* *endl*;

    }

*string* *use*() {

        int i;

        cout *<<* "Which weapon would you like to use? (Enter its number)\n>> ";

        cin *>>* i;

        if (i >= this->weapons.*size*() || i < 0)

            return this->weapons*[*0*]*;

        return this->weapons*[*i*]*;

    }

};

class *Enemy*;

class *Character* : public *Player* {

private:

    int level, points;

*string* experience;

public:

*Character*(int *id*, *string* *name*) : *Player*(*id*, *name*), *level*(0), *points*(0), *experience*("Beginner") {}

    void *levelUp*(int *increment*) {

        if (*increment* < 10) return;

        this->level++;

        this->experience *=* experience\_str[level];

    }

    void *playGame*(*Weapon* *weapons*, *Enemy*& *enemy*);

*string* *getExperience*() { return this->experience; }

};

class *Enemy* : public *Player* {

private:

    int damage;

public:

*Enemy*(int *id*, *string* *name*, int *damage*) : *Player*(*id*, *name*), *damage*(*damage*) {

        if (*damage* > 10 || *damage* < 1)

*damage* = 5;

    }

    void *attacks*(*Weapon* *weapons*, *Character*& *player*) {

        cout *<<* "Enemy\n";

*string* weapon = *weapons*.*use*();

        cout *<<* "Enemy used '" *<<* weapon *<<* "' to attack player\n";

*player*.*setHealth*(*player*.*getHealth*() - damage);

    }

};

void *Character*::*playGame*(*Weapon* *weapons*, *Enemy*& *enemy*) {

    cout *<<* "Character\n";

*string* weapon = *weapons*.*use*();

    cout *<<* "Player used '" *<<* weapon *<<* "' to attack the enemy\n";

*enemy*.*setHealth*(*enemy*.*getHealth*() - 5);

    this->*levelUp*(10);

}

int *main*() {

    cout *<<* "Fasih Hasan Khan\n23k0018\n\n";

*Weapon* *weapons*({

        "Mjoelnir",

        "Funny Sword"

    });

    weapons.*display*();

*Character* *character*(0, "Bob");

*Enemy* *enemy*(1, "Golem", 30);

    for (int i = 0; i < 3; i++) {

        cout *<<* "Character experience: " *<<* character.*getExperience*() *<<* "\n";

        cout *<<* "Character health: " *<<* character.*getHealth*() *<<* "\n";

        character.*playGame*(weapons, enemy);

        cout *<<* "Enemy health: " *<<* enemy.*getHealth*() *<<* "\n";

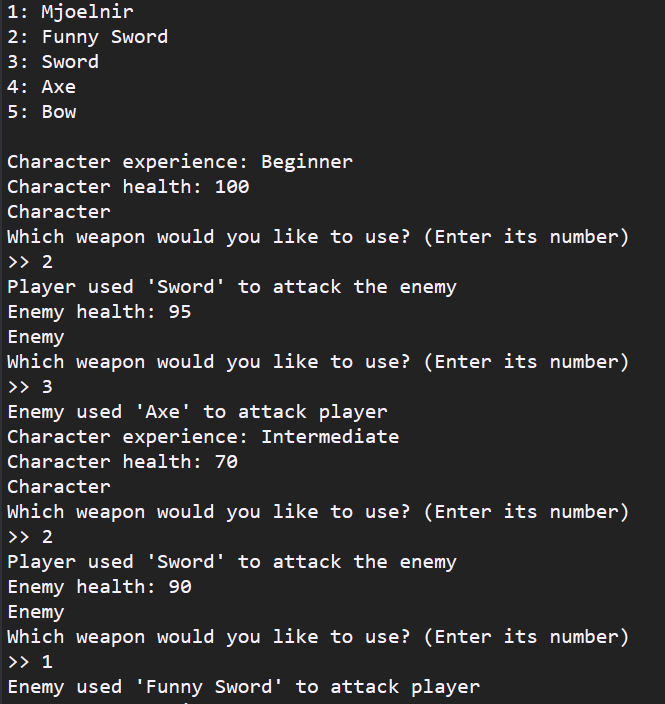
        enemy.*attacks*(weapons, character);

    }

    return 0;

}

**OUTPUT:**

****

**A screenshot of a computer screen

Description automatically generated**

**Q3:**

#include <iostream>

using namespace std;

class *DarazPersonData* {

private:

*string* firstName, lastName, address, city, state, zip, phone;

public:

*DarazPersonData*(*string* *first*, *string* *last*, *string* *addr*, *string* *cty*, *string* *st*, *string* *zp*, *string* *ph*)

        : *firstName*(*first*), *lastName*(*last*), *address*(*addr*), *city*(*cty*), *state*(*st*), *zip*(*zp*), *phone*(*ph*) {}

*string* *getFirstName*() { return this->firstName; }

*string* *getLastName*() { return this->lastName; }

*string* *getAddress*() { return this->address; }

*string* *getCity*() { return this->city; }

*string* *getState*() { return this->state; }

*string* *getZip*() { return this->zip; }

*string* *getPhone*() { return this->phone; }

    void *setFirstName*(*string* *first*) { this->firstName *=* *first*; }

    void *setLastName*(*string* *last*) { this->lastName *=* *last*; }

    void *setAddress*(*string* *addr*) { this->address *=* *addr*; }

    void *setCity*(*string* *cty*) { this->city *=* *cty*; }

    void *setState*(*string* *st*) { this->state *=* *st*; }

    void *setZip*(*string* *zp*) { this->zip *=* *zp*; }

    void *setPhone*(*string* *ph*) { this->phone *=* *ph*; }

};

class *DarazCustomerData* : public *DarazPersonData* {

private:

    int customerNumber, loyaltyPoints;

    static int totalCustomers;

public:

*DarazCustomerData*(*string* *first*, *string* *last*, *string* *addr*, *string* *cty*, *string* *st*, *string* *zp*, *string* *ph*)

        : *DarazPersonData*(*first*, *last*, *addr*, *cty*, *st*, *zp*, *ph*), *customerNumber*(totalCustomers++), *loyaltyPoints*(0) {}

    int *getCustomerNumber*() { return this->customerNumber; }

    int *getLoyaltyPoints*() { return this->loyaltyPoints; }

    void *setCustomerNumber*(int *num*) { this->customerNumber = *num*; }

    void *setLoyaltyPoints*(int *points*) { this->loyaltyPoints = *points*; }

};

int *DarazCustomerData*::totalCustomers = 0;

class *DarazLoyalyProgram* {

private:

*DarazCustomerData* \*customer;

public:

*DarazLoyalyProgram*(*DarazCustomerData* \**customer*) : *customer*(*customer*) {}

    void *addLoyaltyPonts*(int *points*) {

        customer->*setLoyaltyPoints*(customer->*getLoyaltyPoints*() + *points*);

    }

    float *redeemLoyaltyPoints*(int *points*) {

        int loyaly\_points = customer->*getLoyaltyPoints*();

        if (loyaly\_points - *points* < 0 || *points* < 0) return -1.f;

        return 0.75f;

    }

    void *displayLoyaltyPoints*() {

        cout *<<* "Total Loyalty Points: " *<<* customer->*getLoyaltyPoints*() *<<* *endl*;

    }

};

void *print*(*DarazCustomerData* *customer*) {

    cout *<<* "Customer ID: " *<<* *customer*.*getCustomerNumber*() *<<* "\n";

    cout *<<* "Name: " *<<* *customer*.*getFirstName*() *+* " " *+* *customer*.*getLastName*() *<<* "\n";

    cout *<<* "Address: " *<<* *customer*.*getAddress*() *<<* "\n";

    cout *<<* "City: " *<<* *customer*.*getCity*() *<<* "\n";

    cout *<<* "Phone: " *<<* *customer*.*getPhone*() *<<* "\n";

    cout *<<* "Zip: " *<<* *customer*.*getZip*() *<<* "\n";

    cout *<<* "State: " *<<* *customer*.*getState*() *<<* "\n";

}

int *main*() {

    cout *<<* "Fasih Hasan Khan\n23k0018\n\n";

    int loyaltyPoints;

    float item\_price;

*DarazCustomerData* *customer*(

       "Fasih",

       "Khan",

       "Johar",

       "Karachi",

       "State",

       "1001",

       "03XXXXXXX79"

    );

*print*(customer);

*DarazLoyalyProgram* *loyaltyProgram*(&customer);

    loyaltyProgram.*addLoyaltyPonts*(100);

    loyaltyProgram.*displayLoyaltyPoints*();

    cout *<<* "Enter item price: ";

    cin *>>* item\_price;

    cout *<<* "Enter amount of points to redeem: ";

    cin *>>* loyaltyPoints;

    cout *<<* "Original Price: " *<<* item\_price *<<* "\n";

    float discount = loyaltyProgram.*redeemLoyaltyPoints*(loyaltyPoints);

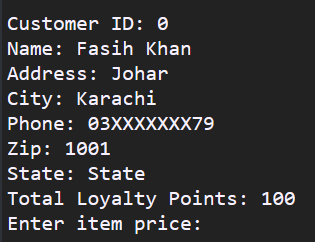
    if (discount == -1) return 1;

    cout *<<* "Discounted Price: " *<<* item\_price \* discount *<<* "\n";

    return 0;

}

**OUTPUT:**

****

**Q4:**

#include <iostream>

#include <algorithm>

using namespace std;

class *Comment* {

private:

    int commentId;

*string* content;

    static int totalComments;

public:

*Comment*(*string* *content* = "")

        : *content*(*content*), *commentId*(totalComments++) {}

    inline void *display*() { cout *<<* "\t[" *<<* commentId *<<* "]: " *<<* content *<<* '\n'; }

};

int *Comment*::totalComments = 0;

class *Post* {

private:

    int postId, likes, views;

*string* content;

*Comment* \*comments;

    int count;

    static int totalPosts;

public:

*Post*(*string* *content* = "")

        : *content*(*content*), *postId*(totalPosts++), *likes*(0), *count*(0), *views*(0), *comments*(**new** *Comment*[10]) {}

    int *getID*() const { return postId; }

    int *getLikes*() const { return likes; }

    int *getViews*() const { return views; }

*Post*& *setLikes*(const int *likes*) {

        this->likes = *likes*;

        return \*this;

    }

*Post*& *setViews*(const int *views*) {

        this->views = *views*;

        return \*this;

    }

*Post*& *addComment*(const *Comment*& *comment*) {

        if (count > 10) return \*this;

        comments[count++] *=* *comment*;

        return \*this;

    }

    inline void *display*() {

        cout *<<* "[" *<<* postId *<<* "]: " *<<* content *<<* '\n';

        cout *<<* "Likes: " *<<* likes *<<* '\n';

        cout *<<* "Views: " *<<* views *<<* '\n';

        cout *<<* "Comments: " *<<* count *<<* '\n';

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

            comments[i].*display*();

        cout *<<* *endl*;

        views++;

    }

};

int *Post*::totalPosts = 0;

void *sortPosts*(*Post* \**posts*, int *count*) {

    std::*sort*(*posts*, *posts*+*count*, [&](const *Post*& *a*, const *Post*& *b*) { return *a*.*getLikes*() > *b*.*getLikes*(); });

    std::*sort*(*posts*, *posts*+*count*, [&](const *Post*& *a*, const *Post*& *b*) { return *a*.*getViews*() > *b*.*getViews*(); });

}

class *User* {

protected:

    char type;

private:

*string* name, email, password;

public:

*User*(*string* *name*, *string* *email*, *string* *password*, char *type*)

        : *name*(*name*), *email*(*email*), *password*(this->*encryptPassword*(*password*)), *type*(*type*) {}

    bool *verifyUser*(const *User*& *user*) { return *user*.name *==* this->name && *user*.email *==* this->email && *user*.password *==* this->password; }

*string* *encryptPassword*(const *string*& *password*) {

*string* new\_password = "";

        for (char c : *password*)

            new\_password *+=* c+1;

        return new\_password;

    }

*string* *getName*() const { return name; }

    void *setName*(const *string*& *newName*) { name *=* *newName*; }

*string* *getEmail*() const { return email; }

    void *setEmail*(const *string*& *newEmail*) { email *=* *newEmail*; }

};

class *RegularUser* : public *User* {

private:

*Post* \*feed;

    int count;

    static const int MAX\_FEED\_SIZE = 10;

public:

*RegularUser*(*string* *name*, *string* *email*, *string* *password*) : *User*(*name*, *email*, *password*, 'r'), *count*(0), *feed*(**new** *Post*[MAX\_FEED\_SIZE]) {}

    void *addToFeed*(const *Post*& *post*) {

        if (count > 5) {

            cout *<<* "Max post limit reached!" *<<* *endl*;

            return;

        }

        feed[count++] *=* *post*;

    }

    void *viewFeed*() const {

        cout *<<* "Feed for " *<<* this->*getName*() *<<* '\n';

        cout *<<* "Posts [" *<<* count *<<* "]\n\n";

        for (short i = 0; i < count; i++)

            feed[i].*display*();

        cout *<<* *endl*;

    }

};

class *BusinessUser* : public *User* {

private:

    int const limit = 10;

    int count;

public:

*BusinessUser*(*string* *name*, *string* *email*, *string* *password*) : *User*(*name*, *email*, *password*, 'b'), *count*(0) {}

    void *promotePost*(const *Post*& *post*, *Post* \**posts*, int *count*) {

        if (type != 'b') return;

        if (this->count > limit) {

            cout *<<* "Post promotion limit reached!" *<<* *endl*;

            return;

        }

        for (int i = 0; i < *count*; i++) {

            if (*posts*[i].*getID*() == *post*.*getID*()) {

                int likes = *post*.*getLikes*(), views = *post*.*getViews*();

*posts*[i].*setLikes*(likes\*2).*setViews*(views\*3);

                break;

            }

        }

*sortPosts*(*posts*, *count*);

        this->count++;

    }

};

int *main*() {

    cout *<<* "Fasih Hasan Khan\n23k0018\n\n";

*Post* posts[] = {

*Post*("Hello World!").*addComment*({ "Hello" }).*addComment*({ "World "}).*addComment*({ "!" }),

*Post*("Promotional Content").*setLikes*(2).*setViews*(1).*addComment*({ "Cool!" }),

*Post*("Interesting").*addComment*({ "Truly" })

    };

*RegularUser* *regular*("Fasih", "fhk", "123");

*BusinessUser* *businesss*("Owais", "ows", "234");

    businesss.*promotePost*(posts[1], posts, 2);

    regular.*addToFeed*(posts[0]);

    regular.*addToFeed*(posts[1]);

    regular.*addToFeed*(posts[2]);

    regular.*viewFeed*();

    return 0;

}

**OUTPUT:**

**A screenshot of a computer

Description automatically generated**