

Assignment # 01

Question 1:

Explain the subsequent questions to each given scenario.

Scenario 1:

A travel agency system is designed to manage flight reservation details for customers. The system needs to store passenger information including:

- ☐ Passenger Name
- ☐ Passport Number
- ☐ Ticket Price
- ☐ Is the Passenger a Senior Citizen
- ☐ Is the Passenger Traveling with Family

The travel agency offers discounts to customers based on certain conditions to promote family-friendly and senior citizen travel.

- ☐ Senior Citizens receive a 20% Discount as part of their loyalty program.
- ☐ Passengers traveling with family receive a 10% Discount to encourage family trips.
- ☐ If both conditions are true, a 30% Discount is applied as a special combined package.

The system must calculate the final ticket price after applying the appropriate discount based on the passenger's eligibility.

- i. **Demonstrate** how if-else if statements along with relational and logical operators are used to calculate the final ticket price.
- ii. **Explain** how boolean variables are utilized to control decision-making during the discount application process.

Pseudocode:

1. Start
2. Declare: Passenger name, Passport number, Ticket price, isSenior & isFamily as Boolean.
3. Print: Enter your name.
4. Input: Passenger name.
5. Print: Enter your Passport number.
6. Input: Passport number.
7. Print: Enter Ticket price.
8. Input: Ticket price.
9. Print: Are you senior?
10. Input: isSenior

11. Print: Are you with your family?
12. Input: isFamily
13. If: isSenior & isFamily are True, then
 Print: You are getting a discount of 30%
14. Else If: isSenior is True, then
 Print: You are getting a discount of 10%
15. Else If: isFamily is True, then
 Print: You are getting a discount of 20%
16. Else:
 Print: Ticket Price
17. End

Code with Explanation:

```
#include<iostream>
#include<string>
using namespace std;
int main(){
    string name;
    int ppn = 0, tp = 0; // ppn -> passport number & tp -> ticket price
    bool isSenior, isFamily;
    cout << "Enter your name: ";
    getline(cin,name);
    cout << "Enter your Passport number: ";
    cin >> ppn;
    cout << "Enter your ticket price: ";
    cin >> tp;
    cout << "Are you senior? (true/false) ";
    cin >> boolalpha >> isSenior; // using 'boolalpha' so that compiler can
    understand true or false as 0 or 1
    cout << "Are you with your family? (true/false) ";
    cin >> boolalpha >> isFamily;
    if(isFamily && isSenior){ // using if-else statement to check wheather he's
    with faimily or not.
        float dp = 0.00;
        dp = tp*0.7;
        cout << "Cogragulations! You are getting a discount of 30%, you dicounted
    price is: "<<dp<<endl;
    }
    else if(isSenior){
        float dp = 0.00; // dp -> discount price
        dp = tp*0.9; // id discount is 10%, it means he's paying 90%
```

```
        cout << "Congratulations! You are getting a discount of 10%, you dicounted  
price is: "<<dp<<endl;  
    }  
    else if(isFamily){  
        float dp = 0.00;  
        dp = tp*0.8;  
        cout << "Congratulations! You are getting a discount of 20%, you dicounted  
price is: "<<dp<<endl;  
    }  
    else{  
        cout << "Your payable amount is: "<<tp<<endl;  
    }  
    return 0;  
}
```

Scenario 2:

A salary management system is developed for a small company to calculate the monthly Net Salary of its employees. The system considers the following salary structure:

- ☐ Basic Salary (user input)
- ☐ House Rent Allowance = 30% of Basic Salary
- ☐ Medical Allowance = 10% of Basic Salary
- ☐ If the employee has been working for more than 10 years, they receive a 500 AED loyalty bonus.
- ☐ If the employee is married, they receive an additional 200 AED Family Allowance.

The company needs to calculate and display the Net Salary of 5 employees. The system should also verify whether each employee is eligible for the loyalty bonus and family allowance.

- i. **Demonstrate** how nested if-else structures are used to apply salary conditions.
- ii. **Construct** the pseudo code using loop to calculate and display the Net Salary for 5 employees.
- iii. **Explain** how the loop iterates and processes employee data efficiently.

Pseudocode:

1. Start
2. Declare: Basic Salary, Rent allowance, Medical allowance, Experience Bonus 500, Married Bonus 200, NetSalary as float, isExperience & isMarried as boolean.
3. Setting 'i' as 1
4. While $i \leq 5 \rightarrow$ Execute
5. Print: Enter your Basic Salary.
6. Input: Basic Salary.
7. Print: Are you experienced?
8. Input: isExperience
9. Print: Are you married?
10. Input: isMarried
11. $\text{Rent allowance} = \text{Basic Salary} \times 0.3$
12. $\text{Medical} = \text{Basic Salary} \times 0.1$
13. $\text{Final Salary} = \text{Basic Salary} + \text{Rent allowance} + \text{Medical allowance}$
14. If: isExperience is True, then

If: isMarried is True, then
Print: Final Salary + Experience Bonus + Married Bonus

Else:
Print: Final Salary + Experience Bonus

15. Else if: isMarried is True, then
 Print: Final Salary + Married Bonus
16. Else:
 Print: Final Salary
17. Add 1 to iterator
18. End

Code with Explanation:

```
#include<iostream>
using namespace std;
int main(){
    int basicsal = 0, i = 1; // basicsal -> basic salary & i -> iterator for 5
employees
    int exp_bonus = 500, married_bonus = 200;
    bool isExp, isMarried; //exp -> experience
    float rentall = 0.00, medall = 0.00, finalsal = 0.00; // rentall -> rent
allowance & medall -> medical allowance
    while(i<=5){ // giving it maximum condition of 5, bcuz we have to calculate
only for 5 employees

        cout << "Enter your basic salary: ";
        cin >> basicsal;
        cout << "Is your experience is more than 10 years? (true/false) ";
        cin >> boolalpha >> isExp;
        cout << "Are you married? (true/false) ";
        cin >> boolalpha >> isMarried;
        rentall = basicsal*0.3;
        medall = basicsal*0.1;
        finalsal = (basicsal + rentall + medall); // doing this here bcuz every
will get minimum salary with rent allowance and medical allowance, even if he has
experience or not also irrespective of his marital status

        if(isExp){ // if employee is experienced there are high chances that he
is married, that is why we are using nested loop here.
            if(isMarried){
                cout << "your net salary is:
"<<finalsal+exp_bonus+married_bonus<<endl;
            }
            else{
                cout << "your net salary is: "<<finalsal+exp_bonus<<endl;
            }
        }
        else if(isMarried){
            cout << "your net salary is: "<<finalsal+married_bonus<<endl;
        }
    }
}
```

```
    }  
    else{  
        cout << "your net salary is: "<<finalsal<<endl;  
    }  
    i++; // increasing it after calculating net salary for each employee  
}  
}
```

Scenario 3:

A student registration system automatically generates a unique registration code for each student based on the following rules:

- ☐ Convert the first letter of the student's name into its ASCII code
- ☐ Append the last two digits of the student's ID to the ASCII code

Sample

Input: Name = **Ali**, ID = **2345**

Output: **A45**

The system should allow multiple student registrations and display the generated registration code for each student.

- i. **Demonstrate** how ASCII conversion is applied using string manipulation functions.
- ii. **Explain** how prefix and postfix increment operators can be used to modify the ASCII code before displaying the result.
- iii. **Construct** the pseudo code to generate the registration code for multiple students using loop.

Pseudocode:

1. Start
2. Declare: Name, ID
3. Print: Enter your name.
4. Input: Name
5. Print: Enter your ID.
6. Input: ID
7. Declare: String ID
8. String ID = Convert ID to String
9. Print: ASCII value of the first character of Name + last two digits of String ID.
10. End

Code + Explanation:

```

#include<iostream>
#include<string>
using namespace std;
int main(){
    string name;
    int id = 0;
    cout << "Enter your name: ";
    getline(cin,name);
    cout << "Enter your ID: ";
    cin >> id;
    string aStr; // here we are declaring a new data type of string, the name
'aStr' is bcuz, we'll convert int 'id' into string for extraction
    aStr = to_string(id); // here we are converting int 'id' into string so that
we can extract digits from it and 'to_string(int_name)' is a function to
convert any int into string
    cout << "Your registration number is:
"<<int(name[0])<<aStr.substr(aStr.length()-2,2)<<endl;
    // int(name[0]) -> it is function to extract ascii value for any charachter
of string, in square brackets 0 indicates that we have to get the ascii value
first character in string.
    // aStr.substr(aStr.length()-2,2) -> as you know that we have already
converted the integer ID to string and the it's new name is aStr, we have done
this so that we can extract digits from
it.
    -> Now, let's understand it, the
(aStr.length()-2,2) the -2 says that we have to go to the second last digit of
our string(as we don't know the lenght of string), and the 2 says that, we have
to extract the two digits.
    -> aStr.subtsr() says that extract
whatever is written in brackets, btw if we know that how many digits are there so
it will be look like this aStr.substr(3,2) 3 indicates go to third place, and and
2 says extract two digits onwards, but we don't know the length of string so that
is why we have written (aStr.length()-2,2) in brackets
}

```


Question 2:

Interpret and Infer the given the task the end of each given scenario.

Scenario 1:

A bank loan repayment system helps customers keep track of their monthly loan installments.

The system follows these rules:

- ☐ Loan Amount = 20,000 AED
- ☐ Monthly Installment = 2000 AED
- ☐ If the customer pays 6 consecutive installments on time, they receive a 1000 AED bonus added to the loan payment.
- ☐ If the customer misses any installment, the bonus is not applied.

The system needs to calculate the remaining loan balance after each payment and stop the process once the entire loan amount is fully paid. The bonus eligibility must be checked after every 6 payments.

- i. **Exhibit** the use of a while loop to simulate the installment payment process.
- ii. **Explain** how the bonus condition is verified inside the loop.
- iii. **Demonstrate** how the remaining loan balance is updated at each iteration and how the loop terminates when the loan is fully repaid.

Pseudocode:

1. Start
2. Declare: loan = 20,000, installment = 2000, counter for consecutive payments, user decision as boolean
3. While loan >= 0 → Execute
 1. Print: Your current loan is 'loan'
 2. Print: Do you want to pay installment?
 3. Input: user decision
 4. If: user decision is True, then
 1. loan = loan - installment
 2. Increase the counter
 3. If: counter = 6, then
 1. loan = loan - 1000
 2. Print: You got a bonus of Rs 1000
 3. counter = 0 // resetting it
 5. Else:
 1. Print: You didn't pay the loan
4. Counter = 0; // Breaking the loop because he doesn't pay loan.
5. While ended.
6. Print: You have successfully paid the loan.

7. End

Code + Explanation:

```
#include<iostream>
using namespace std;
int main(){
    int la = 20000, ma = 2000; // la -> loan amount & ma -> monthly installment
    bool ud; // ud -> user decision
    int count = 0; // counter for consecutive payments
    while(la>=0){
        cout << "Your existing loan amount is: "<<la<<endl;
        cout << "Do you want to pay monthly installment of RS 2000? (0/1) ";
        cin >> ud;
        if(ud){
            la = la-ma;
            count++; // incrementing if he pays the loan
            if(count==6){
                cout << "Congragulations! you got bonus of RS 1000 for paying 6
times consecitively. "<<endl;
                la = la-1000;
                count = 0; // reseting counter when it gets itreates to 6
            }
        }
        else{
            cout << "You didn't pay the loan"<<endl;
            count = 0; // reseting counter to zero if he didn't his streak of
paying loan
        }
    }
    cout << "you have succesfully paid the loan."<<endl;
}
```

Scenario 2:

A number guessing game is designed for entertainment purposes where the system generates a random number between 1 and 50. The player has to guess the number, and the system provides feedback based on the player's input:

- ☐ "Too Low" if the guess is smaller than the number
- ☐ "Too High" if the guess is larger than the number
- ☐ "Correct! You guessed in X attempts" when the guess is correct

The system must count the total number of attempts using the prefix increment operator and display the result when the correct guess is made.

- i. **Demonstrate** the use of do-while loop to repeatedly prompt the player for guesses.
- ii. **Explain** how the prefix operator is used to count the number of attempts.
- iii. **Construct** the pseudo code showing how the loop terminates when the correct guess is made.

Pseudocode:

1. Start
2. Declare: random.no, ID, user.input, iterator for no. of turns: i = 1
3. Do:
 1. Print: A random number is generated, guess it.
 2. Input: user.input
 3. If: user.input < random.no then
 1. Print: Too low
 4. Else if: user.input > random.no then
 1. Print: Too high
 5. Else:
 1. Print: You guessed it correctly in 'i' turns
4. If statement ended.
5. Increment i
6. While: user.input is not equal to random.no.
7. End.

Code + Explanation:

```
#include<iostream>
using namespace std;
int main(){
    int r = 10; // r -> random number -> intialized by programmer
    int user_input = 0;
    int i = 0; // iterator to check no. of turns
    do{ // running it atleast one time if user guess the no. in first turn
        cout << "System has generated a random number between 1 and 50. Guess the
number! ";
        cin >> user_input;
        if(user_input < r ){
            cout << "too low"<<endl;
        }
        else if(user_input > r){
            cout << "too high"<<endl;
        }
        else{
            cout << "you have gussed it correctly! "<<endl;
        }
        i++;
    } while (user_input!=r); // making it to excute until user gussed the right
number
    cout << "gussed in: "<<i<<" turns"<<endl;
}
```