Date: March 9, 2023

Class 2

Lecture Notes

A. Taking Input

1. Taking integer input

```
int x;
cin>>x;
if(x % 2 == 0){
    cout<<x<<" is even";
}
else{
    cout<<x<<" is odd";
}
```

2. Taking float input

```
float f;
cin>>f;

cout<<f<<" x 2 = "<<2 * f;
```

3. Taking character input

```
char c;
cin>>c;
cout<<"The character you entered is "<<c;
```

4. Taking string input

```
string first_name,last_name;
cout<<"Enter your name : ";
cin>>first_name>>last_name;
cout<<"Hello "<<first_name<<" "<<last_name<<" Welcome to NIT-w";
```

6. Taking string input with multiple words

```
string name;
getline(cin,name);
cout<<"Hello "<<name<<" Welcome to this course";
```

7. Taking diff types of input in a single line

```
string stud_name;
float cgpa;
cin>>stud_name>>cgpa;
cout<<"Hello "<<stud_name<<" your CGPA is "<<cgpa;
```

B. Loops

1. while loop

A while loop is a control flow statement used in programming that allows a section of code to be repeatedly executed while a certain condition is true.

```
int count = 11;
while(count <= 10){
    cout<<"Square of "<<count<<" is "<<count * count<<"\n";
    count++;
}
```

2. Infinite loops

An infinite loop is a loop that continues to execute without ever terminating, usually because the loop condition is always true or there is no condition at all

```
string password = "secret";
while (true) {
    string user_password;
    cout << "Enter your password: ";
    cin >> user_password;
    if (user_password == password) {
        cout << "Access granted." << endl;
        break;
    }
    else {
        cout << "Incorrect password. Try again." << endl;
    }
}
```

3. do while loop

Will execute atleast once no matter what the condition is.

```
do{
    cout<<"Square of "<<count<<" is "<<count * count<<"\n";
    count++;
}while(count <= 10);
```

4. for loop

A for loop is a control flow statement used in programming that allows a section of code to be repeatedly executed a specific number of times or for each item in a collection.

```
    for(int i = 1;i <= 10;i++){
        cout<<"Square of "<<i<<" is "<<i * i<<"\n";
    }
```

C. Functions - User Defined

A user-defined function is a function that is created and written by the programmer to perform a specific task or set of tasks. The function can be called from other parts of the program to perform the task whenever required, allowing for code reusability and easier maintenance

```
long long factorial(int x){
    long long f = 1;
    for(long long i = 1;i <= x;i++){
        f = f * i;
    }
    return f;
}

long long combination(int n,int r){
    //  nCr = n! / (r! * (n-r)!)
    return factorial(n)/(factorial(r) * (factorial(n - r)));
}


// function that returns nothing
void welcomeMe(string name){
    cout<<"Hello "<<name<<" Welcome to NIT-W\n";
}
```

D. Functions - Pre-Defined

In programming, pre-defined functions, also known as built-in functions or library functions, are functions that are already implemented in the programming language or in an external library.

1. power

The power function is a pre-defined function that is used to calculate the value of a number raised to a given power.

```
    int x,y;
    cout<<"Enter x and y : ";
    cin>>x>>y;
    int answer = pow(x,y); // x^y i.e. x to the power y
    cout<<"x to the power y : "<<answer<<"\n";
```

2. square root

The sqrt() function is a pre-defined function that is used to calculate the square root of a given number.

```
    int n;
    cout<<"Enter the number : ";
    cin>>n;
    float answer = sqrt(n);
    cout<<"sqrt of "<<n << " is " <<answer<<"\n";
```

3. maximum and minimum

The maximum and minimum functions are pre-defined functions that are used to determine the maximum and minimum values between two or more numbers. These functions are included in the <algorithm> library

```
    int a,b;
    cout<<"Enter two numbers : ";
    cin>>a>>b;
    cout<<"The maximum of these are : "<<max(a,b)<<"\n";
    cout<<"The minimum of these are : "<<min(a,b)<<"\n";
```

4. finding max or min of more than two numbers

```
    cout<<max({1,2,3,4,44,66})<<"\n";
    cout<<min({1,2,3,4,44,66})<<"\n";
```

4. absoulute value

The abs() function is a pre-defined function that is used to calculate the absolute value of a given number

```
    int num;
    cout<<"Enter a number : ";
    cin>>num;
    cout<<"Absolute value of "<<num<<" is  : "<<abs(num);
```