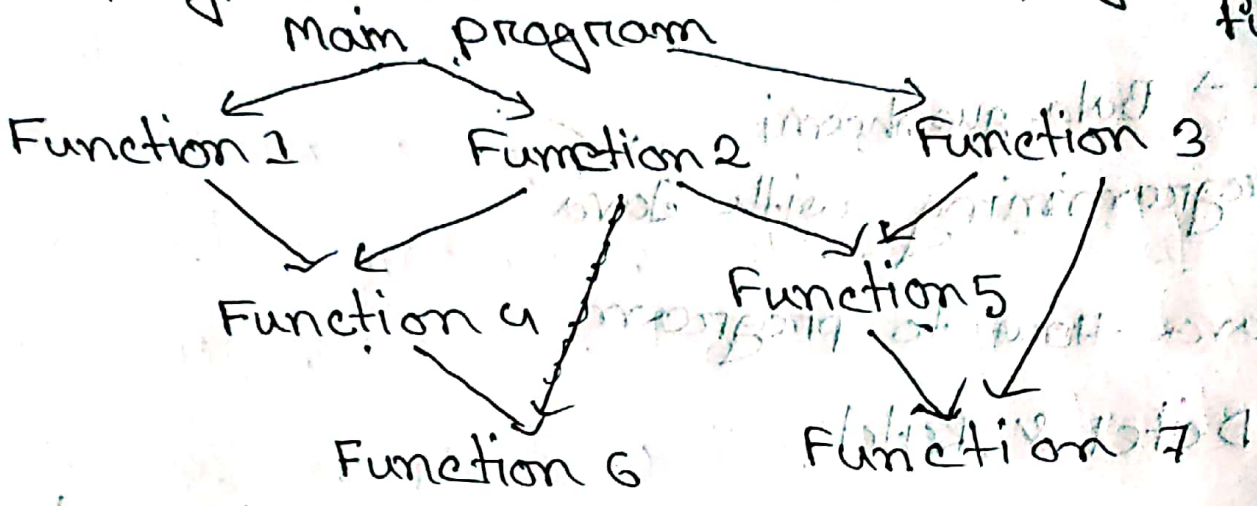
"OOP with Java"

POP/OOP

→ A program in procedural is a set of instruction
→ The program is divided into sub program called function.



What do we have to do to solve this problem.

→ Global ~~val~~ variable.
→ Uncontrolled modification.
→ Problem of enhancement & maintainability
→ Code Reusibility.

Major characteristics of POP:

→ More emphasis is on doing things.
→ It is based on the ~~program~~ problem at hand.

→ used global variable that lead following proplem.

→ It will produce software difficult to maintain.

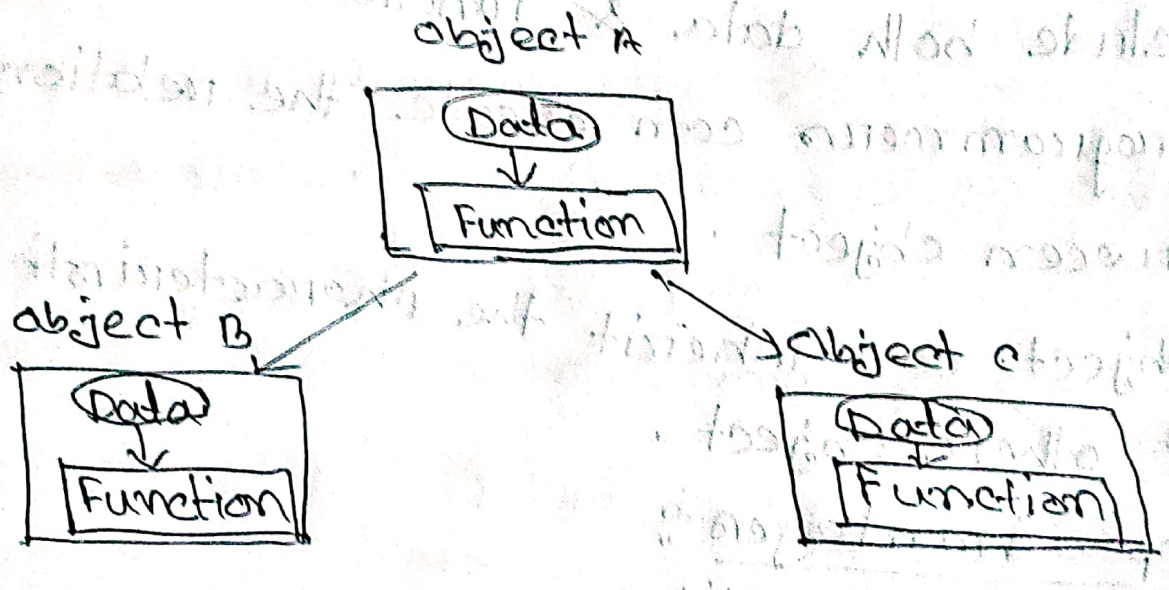
→ The risk of unwanted access & modification increases.

→ It will not result in reusable software.

→ It employs top down approach in program design.

→ Works well in small system.

<u>OOP approach</u>:



OOP:- object oriented programming is a programming paradigm in which program are organized around data or object rather than function & Logic.

→ OOP refers to type of computer programming in which programmer define not only the of data structure but also the types of operation that can be applied to the data structure.

→ The data structure becomes an object that include both data & function.

→ Programmers can create the relationship between object.

→ Object can inherit the characteristics form of other object.

<u>OOP advantages</u>:

→ code reusability.

→ Data Redundancy.

→ code maintanance.

→ Security.

→ Design Benefit.

→ Polymorphism flexibility.

Basic feature of oop :-

→ class
→ object
→ Inheritance
→ Encapsulation
→ Polymorphism
→ Abstruction

Object : object is an instance of class.

→ The object is an entity which has state & behaviour.

→ object has state & behaviour.

State : Represent data (value) of object i.e. balance

behaviour : Represent functionality of an object.

i.e deposit, withdraw.

Example : pen : state : colour, brand, size.

behaviour : writing.

Software : state : data

Class : A class is a blueprint / prototype from which object are created.

→ A class is a blueprint, idea & prototype for creating object.

→ A class is a group of object which have common properties.

Example :- sketch (prototype) of house.

→ It contains all details about floor, door, windows etc.

→ Based on this description we can create house.

→ Here house is the object.

```
class student {
    string name;
    int id;
    int age;
}
```

object:

```
className objectName = new
                        classname();

student obj1 = new student();
obj1.name = "Rahim";
obj1.id = 50;
obj1.age = 20;
```

Encapsulation :-

→ The wrapping up of a data & function into a single unit is called encapsulation.

→ The data is not accessible outside the class.

Abstruction :-

→ ~~Represent the~~ Define representation of essential feature without including background details or explaination.

Inheritance :

→ Refers to a process in which object of one class acquires the property of another class.

Polymorphism :-

→ It means the ability to take more than one form.

Basic feature of Java :

→ Object oriented.
→ compiled and interrepted. (source code → Bite code)
→ Platform independent.
→ Distributed.
→ High performance.
→ simple & familiar.
→ Multithreading.

*

Application:

→ mobile application

→ Desktop " (swing, AWT, Java FX)

→ Web based " (JEE, Servet Let, JSP --)

→ Scientific " (calculator)

→ Gaming "

→ Cloud based "

→ Business "

→ Variable

→ Token

→ Data Type

→ Identifier

→ Operator

→ control statement (if, else, switch)

→ loop statement (for, while, do-while--)

Token: Token is basically smallest element that can identified by the compiler.

Token
- keyword - (int, while)
- identifier - (sum--)
- constant - (10, 20--)
- string - ("Ram", "Rahim")
- special symbol - (), {})
- operator - (+, -, *, /)

identifier: is a sequence of character.
→ are program design token.
→ used for naming class, method, variable, object, package.

keyword: Reserved word / identifier
→ int / float / double, if, switch ----

variable: A variable is an identifier that denotes a storage location used to store data value.
→ Their value changes during the execution of program.
→ Every variable must be declared before used.

Rules: → variable names may consists of alphabet, digit, underscore & dolar sign.
→ They must not begin with a digit.
→ It should not be keyword.
→ ~~Whit~~ uppercase & lowercase are distinct.
→ white space not allowed.
→ variable name can be any length.

Data types :- Data types specifies the size & type of values that can store in variable.

→ primitive → int, char, Boolean, floating

→ Non primitive → class, interface, Array

Operator : An operator is a symbol that tells the computer to perform certain mathematical or logical operation.

→ used in program to manipulate data & variable.

→ Arithmatic operator → (+, -, /, * ...)

→ Relational operator → (==, !=, >, <, >=, <=)

→ logical " → (&&, ||, !)

→ Assignment " → (=, +=, -=, *= ...)

→ Increment / decrement " → (++, --)

→ Conditional " → (? :)

→ Bitwise " → (&, |)

→ Special " → (.)

Program structure :

→ Documentation section
→ Package statement
→ Import statement
→ Interface statement
→ Class Defination
→ Main Method Declaration } Always থাকবে

Package declare না করা হলে default package থাকে।

```
Package // ICT
import Java.io.* ;
Interface → variable → public static Final
          → method → Abstruct method
```

abstruct method এ শুধুমাত্র declaration থাকবে definition থাকবে না।

```
void sum () ;
```

→ class এর নাম ও File এর নাম same হইতে হবে।
→ main Function এর class এর নাম দিয়ে File save করতে হবে।

class name

```
public class Hello {
public static void main (string args [])
{
  System . out . println ("Hello Java");
}
}
```

Class object method

print()
println() → new line create করে.

* The keyword public is an access modifier.

→ The keyword static means that a method is accessable & ~~us~~ usable without object creation.

→ void means main method does not return any value.

* JDK → Java Development kit.

JRE → Java Runtime Environment.

Eclipse IDE – (11)

input নেওয়ার জন্য :

```
import java.util.scanner;
int a;
scanner z = new scanner(system.in);
a = z.nextInt();
```

1.

```
public class Introduction
{
public static void main ( string agrs[] ) {
{
system . out . println ("Nusrat Jahan");
system . out . println ("Akhter Hossain");
system . out . println ("Benu Ara");
system . out . println ("01903549259");
}
}
```

2. Write a program to enter three number & print their sum & average.

```
~~Public class sum {~~
import java . util . scanner ;
Public class sum {
Public static void main (string args[])
{
int a, b, c ;
scanner z = new scanner (system.in);
a = z. nextInt();
b = z. nextInt();
c = z . nextInt();
system . out . println ("Sum =" + (a+b+c));
system . out . println ("Average=" + (a+b+c)/3);
}
}
```

3.

```
import java.util.scanner;
public class max{
public static void main(string args[])
{
int a,b,c;
scanner z = new scanner (system.in);
a = z.nextInt();
b = z.nextInt();
c = z.nextInt();
if (a>b && a>c)
system.out.println("max="+a);
else if (b>a && b>c)
system.out.println("max="+b);
else
system.out.println("max="+c);
}
}
```

4.
```
Import java.util.scanner;
public class OddEven {
public static void main (string arg[])
{
int a;
scanner z = new scanner (system.in);
a = z.nextInt();
if (a%2 == 0)
system.out.println ("Even");
else
system.out.println ("odd");
}
}
```

5.
```
public class Test {
string name;
String dept;
public static void main (string arg[]) {
Test obj = new Test ();
obj.name = "Nusrat";
obj.dept = "ICT";
system.out.println (obj.name);
system.out.println (obj.dept);
}
}
```

```
public class SumAverag {
int average (int a, int b, int c){
return (a+b+c)/3;
}
public static void main (string arg[]) {
int a, b, c;
scanner z = new scanner (system.in);
a = z.nextInt();
b = z.nextInt();
c = z.nextInt();
SumAverage arg = new SumAverage();
System.out.println("Average =", Average (a,b,c));
```

* for loop - 3
  While loop - 3.
  do-while loop - 3
  if-else - 5
  switch - 3
  class - object use করে।

7.
```
Import java.util.scanner;
public class summation {
int sum = 0, i;
for (i=0; i<=n; i++)
{
  sum = sum+i;
}
return sum;
}
public static void main(String arg[]){
int n;
Scanner z = new Scanner (System.in);
n = z.nextInt();
summation.obj=new sum(n);
System.out.println(summation.obj);
}
```

for loop

1. Imparet java.util.scanner;

```
public class summation {
int sum (int[] arr){
int sum = 0;
for (i=0; i<n; i++) {
sum = sum+ arr[i];
}
return sum;
}
}
public static void main (string args[]){
int n, result;
scanner z = new scanner (system.in);
n = z.nextInt();
int[] array = new int[100];
for (int i=0; i<n; i++) {
array [i] = z.nextInt()
}
arraysum.obj = new arraysum();
result = obj.sum (array);
System.out.println(result);
}
}
```

2.
```
import java.util.scanner;
public class Test {
public ~~scanner~~ static scanner z;
int ~~sum~~ evennum(int n) {
int i,n;
for(i=2;i<=n;i=i+2)
{
system.out.println(i);
}
}
public static void main (string arg[]){
int n;
scanner z = new scanner (system.in);
n = z.nextInt();
Test obj = new Test(n);
obj.evennum (n);
}
}
```

3.
```
import java.util.scanner;
public class Test{
public static scanner z;
int odd (int n){
int i, sum=0;
for (i=1, i<=n; i++){
if (n%2!=0)
sum = sum+i;
}
return sum;
}
}
public static void main (string arg[]);
{
int n;
scanner.z = new scanner(system.in);
n=z.nextInt();
Test obj = new Test(n);
system.out.println(obj.odd(n));
}
}
```

I.P – else

```
1. import java.util.scanner;
public class Test {
public static scanner z;
int evenodd (int n){
if (n % 2 ==0)
~~return~~
~~or~~ system.out.println("Even");
else
system.out.println("Odd");
}
}
public static void main (string arg[])
{
int n, ~~result~~;
scanner.z = new scanner(system.in);
n = z.nextInt;
Test obj = new Test(n);
obj.evenodd (n);
}
}
```

2.
```
import java.util.scanner;
public class Test {
public static scanner z;
int num(int n) {
if (n<0)
system.out.println("Negative number");
else if (n>0)
system.out.println("Positive number");
else
system.out.println("zero");
}
public static void main (string arg[]) {
int n;
scanner z = new scanner (system.in);
n = z.nextInt;
Test obj = new Test(n);
obj.num(n);
}
}
```

3.
```
import java.util.scanner;
public class Test {
public static scanner z;
int max (int a, int b, int c) {
if (a>b && a>c)
system.out.println ("Max = "+a);
else if (b>a && b>c)
system.out.println ("Max = "+b);
else
system.out.println ("Max = "+c);
}
public static void main (strint arr[]){
int a, b, c;
scanner.z = new scanner(system.in);
a = z.nextInt();
b = z.nextInt();
c = z.nextInt();
Test obj = new Test ();
obj.Max(a,b,c);
}
}
```

4.
```
import java.util.scanner;
public class Test {
public static scanner z ;
int upperlower (char ch) {
if ( ch >= 'a' && ch <= z')
system.out.println ("uppe lower case");
else
system.out.println ("upper case");
}
}
public static void main (string arg[]) {
char ch ;
scanner z = new scanner (system.in);
ch = z.next().charAt(0);
Test obj = new Test();
obj.upperlower(ch);
}
}
```

5.
```
import java.util.Scanner;
public class Test {
int Leapyear(int y)
{
if (y% 400 == 0)||(y % 100 != 0) && (y % 4 == 0)
system.out.println("Leap year");
else
system.out.println("Not Leap year");
}
}public static void main (String Arg[]){
int y;
scanner input = new scanner (system.in)
y = input.nextInt();
Test obj = new Test();
obj.Leapyear(y);
}
}
```

While loop

1.
```
import java.util.scanner;
public class Test1 {
void num(int n){
int i=1;
while(i<=n)
{
system.out.println(i);
i++;
}
}
}
public static void main(string arg[]){
int n;
Scanner z = new scanner(system.in);
n = z.nextInt();
Test1 obj = new Test1();
obj.num(n);
}
}
```

2.
```
import java.util.scanner;
public class Test2 {
int fact(int n) {
int fact =1;
int i =1;
while (i<=n){
fact = fact *i;
i++;
}
return fact;
}
}
public static void main (string[] ~~ag~~ arg){
int n;
scanner z =new scanner(system.in);
n = z.nextInt();
Test2 obj =new Test2()
system.out.println("Factorial ="obj.fact(n));
}
}
```

3.
```
import java.util.Scanner;
Public class Test3 {
int sum(int n){
int sum = 0, i = 1;
while (i<=n) {
sum = sum+i;
i++;
}
return sum;
}
public static void main(String arr[] ){
int n;
Scanner z = new Scanner(System.in);
n = z.nextInt();
Test3 obj = new Test3()
System.out.println("sum="+obj.sum(n));
}
```

1.
```
import java.util.scanner;
public class Test {
void number(int n){
int i=1;
do{
System.out.print(i);
i++;
}while(i<=n);
}
public static void main(String arr[]){
int n;
Scanner z = new Scanner(System.in);
n = z.nextInt()
Test obj = new Test()
obj.number(n);
}
```

2. ~~Sum of~~

```
import java.util.scanner;
public class display {
void sum (int n) {
int i=1, s=0;
do {
s=s+i;
system.out.println(s);
i++;
}while (i<=n);
}
public static void main (String arc[]){
int n;
scanner z = new Scanner(system.in);
n=z.nextInt();
display obj = new display();
obj.sum(n);
}
```

3.
```
import java.util.scanner;
public class display {
void number (int n) {
int i;
int i=1;
do {
system.out.print("square="+(i*i));
i++
} while (i<=n);
public static void main (string arg[]) {
int n;
scanner z = new scanner(system.in);
n = z.nextInt();
display obj = new display();
obj.number(n);
}
```

Switch

1.
```
import java.util.*;
public class Test {
void months(int n){
case 1 :
system.out.println("January");
break;
case 2 :
system.out.println("February");
break;
case 3:
system.out.println("March");
break;
case 4:
system.out.println("April");
break;
case 5:
system.out.println("may");
break;
case 6:
system.out.println("June");
break;
```

```
case 7:
system.out.println("July");
break;
case 8:
system.out.println("August");
break;
case 9:
system.out.println("September");
break;
case 10:
system.out.println("October");
break;
case 11:
system.out.println("November");
break;
case 12:
system.out.println("December")
break;
default:
system.out.println("Not month");
break;
}
```

```
public static void main (string ar[]){
int n;
scanner z = new scanner (system.in);
n = z.nextInt();
Test obj = new Test();
obj.months();
}
```

2.
```
import.java.util.scanner;
public class test {
void digit (int n){
switch (n){
case 0:
system.out.println("zero");
break;
case 1:
system.out.println("one");
break;
Case 2:
system.out.println("Two");
break;
case 3:
system.out.println("Three");
break;
```

```
case 4:
system.out.println("Four");
break;
case 5:
system.out.println("Five");
break;
case 6:
system.out.println("Six");
break;
case 7:
system.out.println("seven");
break; case 8:
system.out.println("Eight");
break;
case 9:
system.out.println("Nine");
break;
default:
system.out.println("Not a digit"); }
}
}
public static void main(string ar[]);
int n;
scanner z = new scanner(system.in);
n = z.nextInt();
Test obj = new Test();
obj.digit(n);
}
```

5. Print weekdays :

```
import java.util.scanner;
Public class ~~voi~~ Test {
void week (int n){
switch(n){
case 0:
system.out.println ("saturday");
break;
case 1:
system.out.println ("Sunday");
break;
case 2:
system.out.println ("Monday");
break;
case 3:
System.out.println ("Tuesday");
break;
case 4:
system.out.println ("Wednesday");
break;
case 5:
System.out.println ("Thursday");
break;
```

```
case 6:
System.out.println("Friday");
break;
default;
System.out.println("Not a week");
break; }
}
}
public static void main (string ar[]){
int n;
Scanner z=new Scanner(system.in);
n=z.nextInt();
Test ob = new Test();
ob.week(n);
}
```