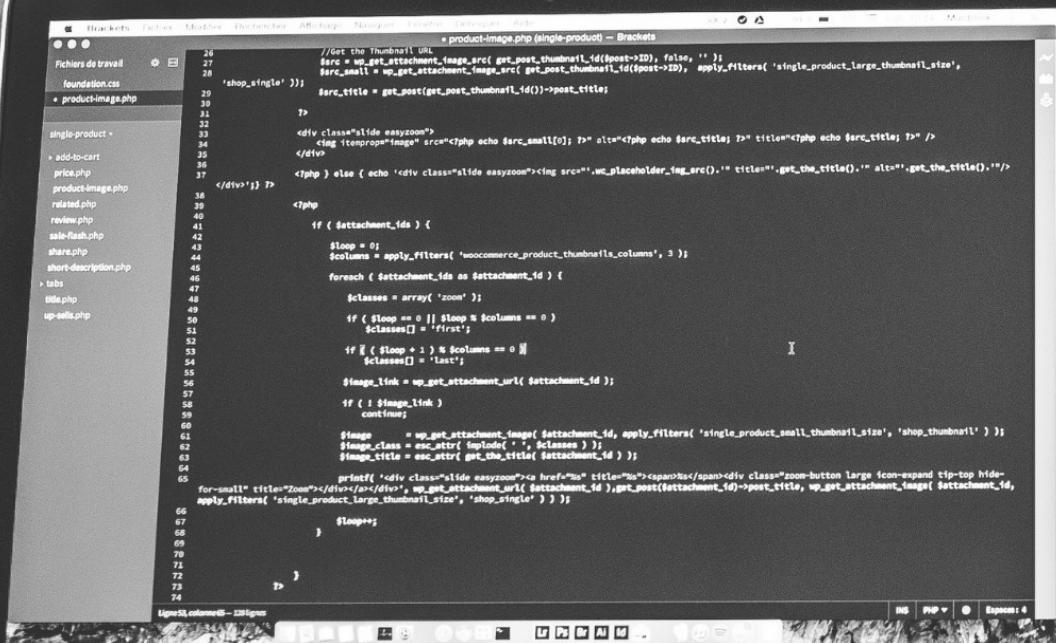


STEP BY STEP

PROGRAMMING IN 15 LANGUAGE



The image shows a laptop screen with a code editor open. The file being edited is 'product-image.php' (single-product). The code is written in PHP and relates to a product image slider. It includes logic for getting attachment URLs, applying filters, and creating a slide structure. The code editor interface shows a sidebar with other files like foundation.css and product-image.php, and a bottom navigation bar.

```
//Get the Thumbnail URL
$src = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), false, '' )
$src_small = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_large_thumbnail_size', 'shop_single' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;

//Get the Small URL
$src_small = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_medium_thumbnail_size', 'shop_thumbnail' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;

//Get the Large URL
$src = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_large_thumbnail_size', 'shop_single' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;

//Get the Zoom URL
$src = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_small_thumbnail_size', 'shop_thumbnail' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;

//Get the Zoom URL
$src = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_medium_thumbnail_size', 'shop_thumbnail' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;

//Get the Zoom URL
$src = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_large_thumbnail_size', 'shop_single' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;

//Get the Zoom URL
$src = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_medium_thumbnail_size', 'shop_thumbnail' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;

//Get the Zoom URL
$src = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_small_thumbnail_size', 'shop_thumbnail' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;

//Get the Zoom URL
$src = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_medium_thumbnail_size', 'shop_thumbnail' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;

//Get the Zoom URL
$src = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_large_thumbnail_size', 'shop_single' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;

//Get the Zoom URL
$src = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_medium_thumbnail_size', 'shop_thumbnail' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;

//Get the Zoom URL
$src = wp_get_attachment_image_src( get_post_thumbnail_id($post->ID), apply_filters( 'single_product_small_thumbnail_size', 'shop_thumbnail' ) );
$post_title = get_post(get_post_thumbnail_id())->post_title;
```

CODING IN DIFFERENT POINTS OF VIEWS
EASY CODE IS PROVIDED IN THIS BOOK

C , C++ , C#(C-SHAP)
JAVA , JAVA SCRIPT
PERL , RUBY , RUST
JULIA , SWIFT , LUA
PHP , GO , PYTHON , R

ENJOY

PROGRAMMING

- LANGUAGE

ABOUT THIS BOOK

MUHAMMAD ALLAH RAKHA

Explanation of programming language has first. Then provided is simple (hello world) in 15 programming language. Later start would programming on 15 language. All context has dependent at coding. I hope you enjoy to this book.

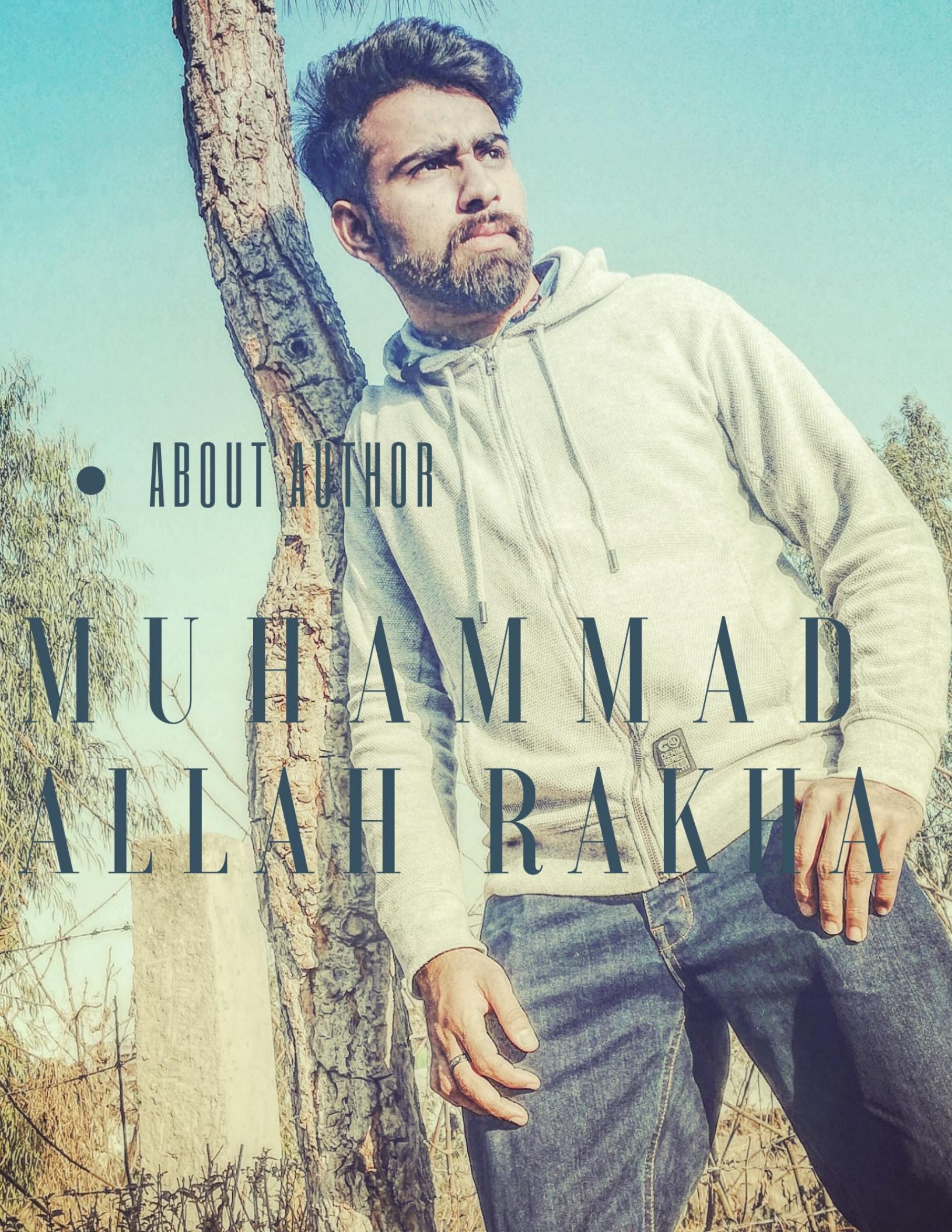
BOOK: PROGRAMMING IN 15 LANGUAGE
AUTHOR: AAAA STARK AND M.A.R
VERSION: 1

AUTHOR IDEA ABOUT THIS BOOK

MUHAMMAD ALLAH RAKHA

This book is easy. In this book has provided a thinking codes. Complex and more complex codes you can solve in this book. So, you can easily improve better coding skills for your self. One time you would solve a complex problem to easy. I hope your mind is creative. Then you can easy read this book start to end.

BOOK: PROGRAMMING IN 15 LANGUAGE
AUTHOR: AAAA STARK AND M.A.R
VERSION: 1



- ABOUT AUTHOR

MUHAMMAD
ALLAH RAKHA

MUHAMMAD ALLAH RAKHA

CO-FOUNDER OF
AAAA STARK INDUSTRIES

MY LIFE

STUDENT: FOUNDATION FOR
ADVANCEMENT OF SCIENCE
ADN TECHNOLOGY UNIVERSITY
SUBJECT: COMPUTER SCIENCE
COUNTRY: PAKISTAN
RELIGION: MUSLIM

LIFE DREAM
SCIENTIST
ADN
FIGHTER



A A A A S T A R K

M Y L O V E

R%oÆŠÀB` `` F ` A ` B ` F 2 ØŒ
LØØX ~ € < F ' ØÆPTTØQØ²ØØÅ =
< F ØØ¹¼Ø = ØÌØ À ` ' Å Q

M E S S A G E

Ø<Ø ØMŒK

Â@ØDÛ|D³Ø}...D²

Ôè@Ø!%ôL

xÿs ^Øn/Ø«ØåïäPØØ ^ÔÒH%uK"óãØØä
ÔTt'6WêØ@.ð™|

ûËsØž·Ú6Ã5XØŸ_Aôh%*RNºl+ØP"3ftºe_
räÃçØi ØÉ‡

**THIS IS ENCRYPTED DATA
HOPE YOU CAN ACCESS**

BASICS IN PROGAMMING LANGUAGE

IN PROGAMMING
BASIC
SYNTEx

IN PROGAMMING
VARIABLE TYPE

IN PROGAMMING
OBJECTS & CLASSES

IN PROGAMMING
MODIFIEER TYPE

IN PROGAMMING
LOOP CONTROL

IN PROGAMMING
DECISION MAKING

IN PROGAMMING
NUMBERS CLASS

IN PROGAMMING
CHARACTER CLASS

IN PROGAMMING
STRING CLASS

IN PROGAMMING
ARRAYS

IN PROGAMMING
RAGULAR EXPRESSIONS

BASICS IN PROGAMMING LANGUAGE

IN PROGAMMING
METHODS

IN PROGAMMING
FILES AND I/O

IN PROGAMMING
EXCEPTIONS

IN PROGAMMING
INNER CLASSES

IN PROGAMMING
INHERITANCE

IN PROGAMMING
OVERRIDING

IN PROGAMMING
POLYMORPHISM

IN PROGAMMING
ABSTRACTION

IN PROGAMMING
ENCAPSULATION

IN PROGAMMING
INTERFACES

HELLO WORLD

IN 15 PROGRAMMING LANGUAGE

```
print("Hello World : Python")
#output: Hello World : Python
```

```
puts("Hello World : Ruby")
#output: Hello World : Ruby
```

```
print("Hello World : Swift")
//output: Hello World : Swift
```

```
print("Hello World : Perl");
#output: Hello World : Perl
```

```
print("Hello World : Lua")
--output: Hello World : Lua
```

```
<?php
    echo ("Hello World : PHP");
    //output: Hello World : PHP
?>
```

```
print("Hello World : R")
#output: Hello World : R
```

```
println("Hello World : Julia ")
#output: Hello World : Julia
```

```
console.log("Hello World : JavaScript")
//output: Hello World : JavaScript
```

```
#include <stdio.h>
int main(){
    printf("Hello World : C ");
    //output: Hello World : C
}
```

```
package main

import "fmt"

func main(){
    fmt.Println("Hello World : GO ")
    //output: Hello World : GO
}
```

```
#include <iostream>
using namespace std;
int main(){
    cout<<"Hello World : C++"<<endl;
    //output: Hello World : C++
    return 0;
}
```

```
import java.util.*;
public class JavaBook{
    // here (JavaBook) your folder name is write.
    Run | Debug
    public static void main(String []args){
        System.out.print("Hello World : Java");
        //output: Hello World : Java
    }
}
```

```
using System;

namespace CSharp{
    class program{
        static void Main(string[] args){
            Console.WriteLine("Hello World : CSharp");
            //output: Hello World : CSharp
        }
    }
}
```

PAGE NUMBER

PYTHON	12
RUBY	22
SWIFT	36
JAVA SCRIPT	43
PERL	50
LUA	58
PHP	66
R	73
RUST	80
GO	86
JULIA	90
CPP	96
C	132
JAVA	134
C# (C SHARP)	153

PYTHON PROGRAMMING LANGUAGE.

MULTI-
PARADIGM: FUNCTIONAL, IMPERATIVE,
OBJECT-
ORIENTED, STRUCTURED, REFLECTIVE

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects

**ABC, ADA, ALGOL 68, APL, C,
C++, CLU, DYLAN, HASKELL,
ICON, JAVA, LISP, MODULA-3,
PERL, STANDARD ML**



NAME

— IN THIS LANGUAGE —

```
print("Muhammad Allah Rakha");
#output: Muhammad ALLah Rakha
name="M.A.R"
print(name)
#output: M.A.R
print(f"My name is {name}")
#output: My name is M.A.R
print("My name is {}".format("AAAA STARK"))
#output: My name is AAAA STARK
print("My name is {}".format(name))
#output: My name is M.A.R
```

PYTHON

PROGRAMMING

```
print("Muhammad Allah Rakha",end=' ')
print("M.A.R") # we use (end=' ') for space
#output: Muhammad Allah Rakha M.A.R
num1=123
print(num1)
#output: 123
print("Number is ",num1)
#output: Number is 123
num2=num1+123456
# here use (+, -, *, %, /, ^)
print(num2)
#output: 123579
print(f"num2 = {num2}")
#output: num2 = 123579
num3=20; name1="AAAA STARK";
name2=f"My name is {name1} and age is {num3}"
print(name2)
#output: My name is AAAA STARK and age is 20
name3="aaaa"; name4="stark";
print(name3+name4)
#output: aaaastark
name5="{} {} {} {}"
print(name5.format("Muhammad","Allah","Rakha","Naich"))
#output: Muhammad Allah Rakha Naich
```

AAAAS T A R K

```
name5="{} {} {} {}"
print(name5.format("Muhammad","Allah","Rakha","Naich"))
#output: Muhammad Allah Rakha Naich

print(name5.format(1,2,3,4))
#output: 1 2 3 4
print(name5.format(1.2,2.3,3.4,4.5))
#output: 1.2 2.3 3.4 4.5
print(name5.format('aaaastark','M','A','R'))
#output: aaaastark M A R
name6="""Muhammad
| Allah Rakha
| | | | Naich"""
print(name6)
#output: Muhammad
#        Allah Rakha
#                Naich
print(f"My name is {name6}")
#output: My name is Muhammad
#        Allah Rakha
#                Naich
print ("Enter your Name")
name1=input()
#enter: AAAA STARK
print ("Enter your Age")
num1=input()
#enter: 20
print (f"Your name is {name1} and age is {num1}")
#output: Your name is AAAA STARK and age is 20
name2=input("Enter your Name.")
#enter: AAAA STARK
num2=input("Enter your Age.")
#enter: 20
print(f"Name is {name2} and age is {num2}")
#output: Name is AAAA STARK and age is 20
import math
name1=abs(-12345)
print(name1) # abs (interger)
#output: 12345
name2=math.ceil(12.345)
print(name2) # ceiling values
#output: 13
name3=math.exp(12.5)
print(name3) # exponential
#output: 268337.2865208745
name4=math.fabs(-123.4)
print(name4) # fabs (float)
#output: 123.4
name5=math.floor(12.999)
print(name5) # floor
#output: 12
name6=math.log(11.22)
print(name6) # Logarithm
#output: 2.4176979000945504
```

BOOK OF CODING

AAAASSTARK

```
name7=math.log10(44.55)
print(name7) # Logarithm (base-10)
#output: 1.6488477083728936
name8=math.modf(12.34)
print(name8) # modf (fractional&interger)
#output: (0.33999999999999986, 12.0)
name9=math.pow(2,2)
print(name9) # power
#output: 4.0
print(int(name9))
#output: 4
name10=math.sqrt(3)
print(name10) # square root
#output: 1.7320508075688772
name11=max(11,22,33)
print(name11) # maximum number
#output: 33
name12=min(11,22,33)
print(name12) # minimum number
#output: 11
num1=10
num2=15
if((num1==num2)and(num1<num2)):
    print("1: Work")
else:
    print("2: Not Work")
#output: 2: Not Work
if((num1!=num2)or(num1<num2)):
    print("1: Work")
else:
    print("2: Not Work")
#output: 1: Work
if(not(num1<num2)):
    # this condition is false
    print("1: Work")
else:
    print("2: Not Work")
#output: 2: Not Work
name="AAAA STARK"
if(name=="aaaa"):
    print(f"1: {name}")
elif(name=="aaaastark"):
    print(f"2: {name}")
elif(name=="stark"):
    print(f"3: {name}")
else:
    print(f"4: {name}")
#output: AAAA STARK
name1=[1,2,3,4,5]
for i in name1:
    print(f"{i} ")
#output: 1 2 3 4 5
name2=["Muhammad","Allah","Rakha"]
for anyone in name2:
    print(anyone)
#output: Muhammad Allah Rakha
```

```
num1=1
name3=[]
while num1<=5:
    print(num1,end=' ')
    name3="M.A.R"
    # here index not allow
    # name3[num1] not allow
    num1=num1+1
    print(name3)
    # print(name3[num1]) not allow
#output: 1 M.A.R
#        2 M.A.R
#        3 M.A.R
#        4 M.A.R
#        5 M.A.R

name1=['M','A','R']
print(name1[0],name1[1],name1[2])
#output: M A R
name2={"N1":123,'N2':"M.A.R","1":12.345}
print(name2["N1"],name2["N2"],name2["1"])
#output: 123 M.A.R 12.345
del name2['N2']
# here delete commad use
# now 'N2' is delete.
print(name2['N2'])
#output: KeyError: 'N2'

#FUNCTIONS
def name1():
    print("M.A.R")
name1()
#output: M.A.R
def name2():
    print("M.A.R")
if __name__ == "__main__":
    name2()
#output: M.A.R

def name1():
    print("Muhammad Allah Rakha")
name1()
#output: Muhammad Allah Rakha
def name2():
    print("Muhammad Allah Rakha");
name2(); # No differences in ();
#output: Muhammad Allah Rakha
def name3(name):
    print(f"Name is {name}")
name3("M.A.R")
#output: Name is M.A.R
def name4(name,num):
    print(f"Name: {name} Age: {num}")
name4("AAAA STARK",20)
#output: Name: AAAA STARK Age: 20
```

BOOK OF CODING

AIAA STARK

```
def name5(*names):
    n1,n2,n3=names
    print(f"Name : {n1}.{n2}.{n3}")
name5('M','A','R')
#output: Name : M.A.R
#     # Same type Function Name use
#     # but values is different now
#     # You can use same or different
def name5(*names):
    n1,n2,n3=names
    print(f"Name is {n1} {n2} {n3}")
name5('Muhammad','Allah','Rakha')
#output: Name is Muhammad Allah Rakha

def names(num,name):
    print(f"{num} : {name}")
num1=1
name1="Muhammad"
names(num1,name1)
#output: 1 : Muhammad
num1=num1+1
name1="Allah"
names(num1,name1)
#output: 2 : Allah
num1=num1+1
name1="Rakha"
names(num1,name1)
#output: 3 : Rakha
def names(name1,name2):
    print(f"N1:{name1} N2:{name2}")
    #output: N1:AAAA N2:STARK
    return name1+name2
    # one string_array_name is simple return
    # but two string_array_name then is (+)
return_name=names("AAAA","STARK")
print(return_name)
#output: AAAASTARK

def number(num1,num2):
    print(f"N1:{num1} N2:{num2}")
    #output: N1:5 N2:5
    return num1+num2
    # return_type you can use
    # (+,-,*,%,/,{^}) anyone
return_number=number(5,5)
print(return_number)
#output: 10
def names_3():
    n1="Muhammad";
    n2="Allah";
    n3="Rakha";
    return n1,n2,n3;
N1,N2,N3=names_3()
print(f"N1:{N1} N2:{N2} N3:{N3}")
#output: N1:Muhammad N2:Allah N3:Rakha
```

AIAA STARK BOOK OF CODING

```
name="Muhammad Allah Rakha";
def name1(n):
    print(f"Name is {n}")
def name2():
    name1
    # name1 function is call now here
name1(name)
name2()
#output: Name is Muhammad ALLah Rakha
# My text_file data (stark.txt)
# Muhammad Allah Rakha
# Nacih.And AAAA STARK
name1=open("stark.txt",'r')
# 'r' use for reading mode
print(name1.read())
#output: Muhammad Allah Rakha
#      Nacih.And AAAA STARK

name1=open("stark.txt",'w')
# 'w' use for writting mode
print("Enter anyone for file_write!")
#output: Enter anyone for file_write!
name2="Jahangir Akhter"
name3="Kousar Perveen"
name4="Hashim Raza"
name5="Muhammad Allah Rakha"
# get_starting writting
name1.write(name2)
name1.write("\n")
#write: Jahangir Akhter
name1.write(name3)
name1.write("\n")
#write: Kousar Perveen
name1.write(name4)
name1.write("\n")
#write: Hashim Raza
name1.write(name5)
#write: Muhammad Allah Rakha
print("End of file to be Close.")
#output: End of file to be Close.
name1.close()

#get_starting reading
name6=open("stark.txt",'r')
print(name6.read())
#output: Jahangir Akhter
#      Kousar Perveen
#      Hashim Raza
#      Muhammad ALLah Rakha
```

BOOK OF CODING

AAAASSTARK

```
#CLASS, OBJECTS
from sys import exit
from random import randint
from textwrap import dedent

class Name(object):
    # object is must or not must
    # depend on Data you enter.
    def display1(self):
        print("Hashim Raza")
    def display2(self):
        print("Hassan Raza")
    # (self) is must when you create
    # class with (object) for Function.
    # and (self) also must in which Function
name=Name()
name.display1()
#output: Hashim Raza
name.display2()
#output: Hassan Raza

class Name1(object):
    def __init__(self, name):
        self.name=name

    def display(self):
        print("M.A.R")
        print (f"{self.name}")

name1=Name1("AAAA STARK")
name1.display()
#output: M.A.R
#output: AAAA STARK
class Name2():
    def display3(self):
        print("aaaastark")
    # (self) also must in which Function
name2=Name2()
name2.display3()
#output: aaaastark

class Name3(object):
    def __init__(self, names,age):
        self.names=names
        self.age=age
    def display4(self):
        # here show that function
        # is also (self) type.
        print("Wellcome: AAAA STARK")
        print(f"Your name is {self.names}")
        print(f"Your age is {self.age}")
name3=Name3("M.A.R",20)
name3.display4()
#output: Wellcome: AAAA STARK
#           Your name is M.A.R
#           Your age is 20
```

BOOK OF CODING

AAAAS TARK

```
# INHERITANCE

class Name4(object):
    def __init__(self, names,age):
        self.names=names
        self.age=age
    def display5(self):
        # here show that function
        # is also (self) type.
        print("Wellcome: AAAA STARK")
        print(f"Your name is {self.names}")
        print(f"Your age is {self.age}")

class Name5(Name4):
    pass
    # this (pass) is defined here (Name4)
    # class exicute.
    # here (display5()) use for both
name4=Name4("M.A.R",20)
name4.display5()
#output: Wellcome: AAAA STARK
#           Your name is M.A.R
#           Your age is 20
name5=Name5("AAAA STARK",20)
name5.display5()
#output: Wellcome: AAAA STARK
#           Your name is AAAA STARK
#           Your age is 20

class Name6(object):
    def display6(self):
        print("M.A.R")
class Name7(Name6):
    def display6(self):
        print("AAAA STARK")
name6=Name6()
name6.display6()
#output: M.A.R
name7=Name7()
name7.display6()
#output: AAAA STARK
class Name8(object):
    def display7(self):
        print("M.A.R")
class Name9(Name8):
    def display7(self):
        print("AAAA STARK")
        # this (super) is call (Name8)
        # of (display7(self)) data.
        super(Name9,self).display7()
name8=Name8()
name8.display7()
#output: M.A.R
name9=Name9()
name9.display7()
#output: AAAA STARK
#output: M.A.R
```

AAAASSTAAARK

BOOK OF CODING

#CALENDER

```
import calendar  
name1=calendar.month(2020,9)  
print("This Month Calender.")  
print(name1)  
#output: This Month Calender.  
# September 2020  
# Mo Tu We Th Fr Sa Su  
# 1 2 3 4 5 6  
# 7 8 9 10 11 12 13  
# 14 15 16 17 18 19 20  
# 21 22 23 24 25 26 27  
# 28 29 30
```

#TIME

```
import time  
print ("Time: ",time.ctime())  
name1=time.localtime()  
print("Time.Localtime is : ",name1)  
#output: Time.Localtime is :  
# time.struct_time(tm_year=2020,  
# tm_mon=9, tm_mday=19, tm_hour=11,  
# tm_min=24, tm_sec=58, tm_wday=5,  
# tm_yday=263, tm_isdst=0)
```

RUBY PROGRAMMING LANGUAGE.

MULTI-
PARADIGM: FUNCTIONAL, IMPERATIVE,
OBJECT-ORIENTED, REFLECTIVE

Ruby is an interpreted, high-level, general-purpose programming language. It was designed and developed in the mid-1990s by Yukihiro "Matz" Matsumoto in Japan. Ruby is dynamically typed and uses garbage collection. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. According to the creator, Ruby was influenced by Perl, Smalltalk, Eiffel, Ada, Basic, and Lisp.

CLOJURE, COFFEESCRIPT, CRYSTAL, D, ELIXIR, GROOVY, IOKÉ, JULIA, MIRAH, NU, RING, RUST, SWIFT

NAME

IN THIS LANGUAGE



```
puts ("Muhammad Allah Rakha")
#output: Muhammad ALLah Rakha
p ("AAAAA STARK")
#output: "AAAAA STARK"
```

RUBY

PROGRAMMING

```
# INTEGERS
x=10
puts(x)
# output: 10
puts 10/3
#output: 3
num1=100
num2=444
num3=num1+num2
puts(num3)
# output: 544
age=20
puts "My age is 20." if age==20
#output: My age is 20

# LOOPING NUMBERS INTEGERS AND STRING
5.times{puts "Muhammad Allah Rakha"}
#output: Muhammad ALLah Rakha
# Muhammad ALLah Rakha
# Muhammad ALLah Rakha
# Muhammad ALLah Rakha
# Muhammad ALLah Rakha
1.upto(5){|number| puts number}
#output: 1 2 3 4 5

10.downto(1){|number1| puts number1}
#output: 10 9 8 7 6 5 4 3 2 1
```

AAAAA STARK
BOOK OF CODING

AAAASSTARK

```
# FLOATING POINT NUMBERS
puts 10.0/3.0
#output: 3.3333333333333335

x=11.0
y=5.0
puts x.to_f/y.to_f
# to_f is define a float value.
#output: 2.2

u=33.222
puts u.to_i
# to_i is define a int value.
#output 33

# STRING AND TEXT
puts "AAAA STARK"
#output: AAAA STARK

x="AAAA"
y=" STARK"
puts "You right!" if x+y=="AAAA STARK"
#output: You right!

# Types of maximum text write
# 1_type for you
x=%q{My name is Muhammad
      Allah Rakha.I am from
      Multan.And me study of
      computer science.
}
puts x
#output: My name is Muhammad
#      Allah Rakha.I am from
#      Multan.And me study of
#      computer science.

# 2_type for you
y=%q!My name is Muhammad
Allah Rakha.I am from
Multan.And me study of
computer science.!
puts y
#output: My name is Muhammad
#      Allah Rakha.I am from
#      Multan.And me study of
#      computer science.

# 3_type for you
z=<<When_you_start_string_with_variable_name_type
My name is Muhammad
Allah Rakha.I am from
Multan.And me study of
computer science.
When you start string with variable name type
puts z # when you starting and ending is same_name_concept
#output: My name is Muhammad
#      Allah Rakha.I am from
#      Multan.And me study of
#      computer science.
```

BOOK OF CODING

```
name="Muhammad Allah Rakha"
puts_(name+" "+name)
#output: Muhammad Allah Rakha Muhammad Allah Rakha
puts_(name+name)
#output: Muhammad Allah RakhaMuhammad Allah Rakha
puts_(name.capitalize())
#output: Muhammad allah rakha
puts_(name.downcase())
#output: muhammad allah rakha
puts_(name.chop())
#output: Muhammad Allah Rakh
puts(name.next())
#output: Muhammad Allah Rakhb
puts_(name.reverse())
#output: ahkaR hALLA dammahuM
puts_(name.reverse.next())
#output: ahkaR hALLA dammahuN
puts_(name.sum())
#output: 1843
puts_(name.swapcase())
#output: mUHAMMAD aLLAH rAKHA
puts_(name.upcase())
#output: MUHAMMAD ALLAH RAKHA
puts_(name.upcase.reverse())
#output: AHKAR HALLA DAMMAHUM
puts_(name.downcase.reverse())
#output: ahkar halla dammahum
# ASCII and CHARACTER
x="A"
puts x.ord_
# ord is use for ASCII value
#output: 65
puts 65.chr
# chr is use ASCII TO CHARACTER value
#output: A
# Interpolation
x=10
y=20
puts "#{x} + #{y} = #{x+y}"
# any arithmetic operation
#/ * + - ^
#output: 10 + 20 = 30
num1=333
num2=444
puts(num1.to_s + " + " + num2.to_s + " = " + (num1+num2).to_s)
puts("#{num1} + #{num2} = #{num1+num2}")
#output: 333 + 444 = 777
# 333 + 444 = 777
name1="Muhammad Allah Rakha Naich"
puts(name1)
puts "\t" # two types of puts.
puts("#{name1}")
#output: Muhammad Allah Rakha Naich
# Muhammad Allah Rakha Naich
```



```

name2="Me print my name is 4 time #{name1*4}"
puts(name2)
#output: Me print my name is 4 time
#           Muhammad ALLah Rakha Naich_
#           Muhammad ALLah Rakha Naich
#           Muhammad ALLah Rakha Naich
#           Muhammad ALLah Rakha Naich

# | | | | #SUBSTITUTIONS
name="Muhammad Allah Rakha Naich"
# 1 way of substitution

puts(name.sub("Naich","Doon"))
# .sub is use for find word(string) then replace anyone.
#output: Muhammad ALLah Rakha Doon
puts(name.gsub('A','a'))
# .gsub is use for find word(character) then replace anyone.
#output: Muhammad allah Rakha Naich

# 2 way of substitution

puts(name.sub(/^\...../,"MUHAMMAD"))
# starting name(string) to change anyone word
# that's 8(dots) is represent of (Muhammad)
#output: MUHAMMAD ALLah Rakha Naich
puts(name.sub(/\....$/,"NAICH"))
# ending name(string) to change anyone word
# that's 5(dots) is represent of (Nacih)
#output: Muhammad ALLah Rakha NAICH

# | | | | #ITERATION WITH A REGULAR EXPRESSION
name="aaaastark"
name1="This is a AAAASTARK"
# .scan is used for scanning total string
# at length then print blook shape.
name.scan(/./){|anyone| puts anyone}
# 1(dots) is represent of 1(character)
# that scan of one by one character
# and print blook Looking shape
#output: a
#       a
#       a
#       s
#       t
#       a
#       r
#       k

name.scan(/\../{|anyone| puts anyone}
# 2(dots) is represent of 2(character)
# that scan of tow by tow character
# and print blook Looking shape
#output: aa
#       aa
#       st
#       ar

# Last character is (k) a miss
# therefor combination is not
# math of another character

```

```
name1.scan(/\\w\\w/){|anyone| puts anyone}
# 2(dots) is represent of 2(character)
# that scan of tow by tow character in
# which miss one single character
# and print blook Looking shape
# output: Th
#         is
#         is
#         AA
#         AA
#         ST
#         AR
#         # Last character is (K) a miss
#         # therefor combination is not
#         # math of another character

name.scan(/[aeiou]/){|anyname| puts anyname}
# scanning of Vowel Words in this
# string at Length
# output: a
#         a
#         a
#         a
#         # that (stark) word not a
#         # print because this is not
#         # Vowel count
puts "Yes Vowel is match in this Name" if name=~/[aeiou]/
# output: Yes Vowel is match in this Name

name.scan(/[a-z]/){|anyname| puts anyname}
# scanning of a_to_z Words of abcde(Language) string
# output: a
#         a
#         a
#         a
#         s
#         t
#         a
#         r
#         k
#         #ARRAY AND LISTS
array=[1,2,3,4,5]
puts(array)
#output: 1 2 3 4 5
puts("Array[0] =#{array[0]} and Array[4] =#{array[4]}")
#output: Array[0] =1 and Array[4] =5
array1=["Muhammad","Allah","Rakha"]
puts(array1)
#output: Muhammad
#         Allah
#         Rakha
array1[2]=" Naich"
puts("Array1[2] is add of #{array1[2]}")
#output: Array1[2] is add of Naich
```

BOOK OF CODING

AAAAA STARK

```
x=[]
x<<"AAAAA_STARK"
x<<"INDUSTRY"
puts_( "We add name in (x)array type")
puts_(x)
#output: We add name in (x)array type
#        AAAA_STARK
#        INDUSTRY

y=[]
# you use (integer,string,float,etc.)
# in array of (y)
y.push(1)
y.push(2)
y.push("Number")
puts(y)
# print of array in first to end
#output: 1 2 Number
puts_(y.pop)
puts_(y.pop)
puts_(y.pop)
# .pop is print of array in reverse order.
# end to first
#output: Number 2 1

array=["Muhammd", "Allah", "Rakha"]
puts_(array.join('.'))
puts_(array.join('.'))
puts_(array.join('.'))
puts_(array.join())
#output: Muhammd.Allah.Rakha
#        Muhammd.Allah.Rakha
#        Muhammd_Allah_Rakha
#        MuhammdAllahRakha

name="aaaa_stark"
puts_(name.scan(/\w/).join('.'))
#output: a,a,a,a,s,t,a,r,k
name1="Muhammad. Allah. Rakha"
let1=(name1.split(/\./).inspect)
let2=(name1.split(/\./))
puts_(let1)
#output: ["Muhammad", " Allah", " Rakha"]
puts_(let2)
#output: Muhammad
#        Allah
#        Rakha
name2="Muhammad Allah Rakha Naich"
puts_(name2.split(/\s+/))
#output: Muhammad
#        Allah
#        Rakha
#        Naich
name_of_array=["Muhammad", "Allah", "Rakha", "Naich"]
name_of_array.each{|anyone|puts anyone}
#output: Muhammad
#        Allah
#        Rakha
#        Naich
```

BOOK OF CODING

```
name_of_array.each{|anyname| puts anyname.to_s}
#output: Muhammad
#        Allah
#        Rakha
#        Naich
num_of_array=[1,2,3,4,5,6,7,8,9,10]
num_of_array.collect{|anyname| puts anyname*5}
#output: 5 10 15 20 25 30 35 40 45 50

x=[1,2,3,4,5]
y=['x','y','z']
name=x+y
puts(name)
#output: 1 2 3 4 5 x y z
puts_(x.include?(1))
#output: true
puts_(y.include?('x'))
#output: true
puts_(x.first)
#output: 1
puts_(x.last)
#output: 5
puts_(x.first(5).join('-'))
#output: 1-2-3-4-5
puts_(y.reverse)
#output: z y x
#HASHES
name_of_dictionary={'Me'=>"Muhammad Allah Rakha",'Comp'=>"AAAA STARK"}
puts_(name_of_dictionary['Me'])
#output: Muhammad Allah Rakha
puts_(name_of_dictionary['Comp'])
#output: AAAA STARK
name_of_dictionary['Comp']="AAAA STARK INDUSTRY"
puts_(name_of_dictionary['Comp'])
#output: AAAA STARK INDUSTRY

name_of_dictionary.each{|key,value| puts "#{key} and #{value}"}
#Hash Elements
#output: Me and Muhammad Allah Rakha
#        Comp and AAAA STARK INDUSTRY
let1=name_of_dictionary.keys
puts_(let1) # retrieving keys
#output: Me
#        Comp
let2=name_of_dictionary.values
puts_(let2) # retrieving values
#output: Muhammad Allah Rakha
#        AAAA STARK INDUSTRY
let3=name_of_dictionary.delete('Me')
puts_(name_of_dictionary)
# when me delete of key "Me" then array in which only one Key "Comp".
#output: {"Comp"=>"AAAA STARK INDUSTRY"}
```

BOOK OF CODING

```
name={  
    "let1"=>{  
        "Me1"=>"Muhammad",  
        "Me2"=>"Allah",  
        "Me3"=>"Rakha",  
        "MeAll"=>["Muhammad", "Allah", "Rakha"]  
    },  
    "let2"=>{  
        "Me4"=>"AAAA",  
        "Me5"=>"STARK"  
    }  
}  
puts_(name["let1"]["Me1"])  
#output: Muhammad  
puts_(name["let1"]["Me2"])  
#output: Allah  
puts_(name["let1"]["Me3"])  
#output: Rakha  
puts_(name["let1"]["MeAll"])  
#output: Muhammad Allah Rakha  
puts_(name["let1"]["MeAll"].join(','))  
#output: Muhammad,Allah,Rakha  
puts_(name["let2"]["Me4"])  
#output: AAAA  
puts_(name["let2"]["Me5"])  
#output: STARK  
  
#FLOW CONTROL (CONDITIONS) IF, ELSE, ELSEIF, etc  
me_age=20  
if(me_age)<25  
    puts "You are young man"  
else  
    puts "You are teenage man"  
end  
#output: You are young man  
you_age=20  
if(you_age)>25  
    puts "Your age is less of 25"  
elsif(you_age)<22  
    puts "You are teenage man"  
end  
#output: You are teenage man  
#Ternary Operator  
name="aaaastark"  
let1=name=="aaaastark" ? "AAAA STARK" : "M.A.R"  
puts "Wellcom to "+let1  
#output: Wellcom to AAAA STARK  
  
#while  
1.upto(5){|anyname|puts anyname}  
#output: 1 2 3 4 5  
x=1  
while x<=4  
    puts x  
    x=x+1  
end  
#output: 1 2 3 4
```

BOOK OF CODING

AAAASSTARK

```
#code bloack
u=[1,2,3]
u.each{|u| puts u}
#output: 1 2 3

#FUNCTIONS
# Local variables
def name_any()
  x=10
  puts x
end
name_any()
#output: 10
x=15
puts x
#output: 15

#global variables
def name_any_one()
  puts $name
end
$name="AAAA STARK"
name_any_one()
#output: AAAA STARK

def add_two_num(num1,num2)
  puts num1+num2
end
add_two_num(22,22)
#output: 44

def name_two(nam1,nam2)
  return nam1+nam2
end
let1=name_two("AAAA ","STARK")
puts (let1)
#output: AAAA STARK
def just_free()
  i=0
  o=i+1
  puts (o)
end
just_free()
#output: 1

#CLASSES,OBJECTS
x="AAAA STARK"
puts x.class
#output: String
puts x.downcase
#output: aaaa stark
puts x.length
#output: 10
```

BOOK OF CODING

AAAAA STARK

```
class Name_1
  attr_accessor :name,:num
end
let1=Name_1.new
let1.name="AAAA STARK"
let1.num=12345
puts let1.name
#output: AAAA STARK
puts let1.num
#output: 12345
puts let1.instance_variables
#output: @name
#           @num
# Public Methods
class Names
  def initialize(anyone)
    @anyone=anyone
  end
  def return_type()
    return @anyone
  end
end
let1=Names.new("M.A.R")
names=let1.return_type()
puts names
#output: M.A.R
class Name # class object of
  # first_letter is capital
  def initialize(my_choice)
    # this (initialize) is bulit-type
    @my_choice=my_choice
    # this @ used in class (initialize)-type
  end
  def display_my_choice_value()
    puts @my_choice
  end
end
let1=Name.new("AAAA STARK")
let2=Name.new("Muhammad Allah Rakha")
let3=Name.new(12345)
puts let1.display_my_choice_value()
#output: AAAA STARK
puts let2.display_my_choice_value()
#output: Muhammad Allah Rakha
puts let3.display_my_choice_value()
#output: 12345
```

BOOK OF CODING

AAAAA STARK

```
# Private Methods
class Me1
  def initialize(name)
    @name=name
  end
  def display_name_age()
    age1=age # age() call
    puts "Name is #{@name}"
    puts "and age is #{age1}"
  end
  private
    def age()
      return 20
    end
end
let1=Me1.new("AAAA STARK")
let1.display_name_age()
#output: Name is AAAA STARK
#           and age is 20
#INHERITANCE CLASSES
class Me1
  def name_1()
    puts "AAAA"
  end
  def name_2()
    puts "STARK"
  end
end
class Me2 < Me1
  def name_2()
    # same function use in inheritance
    puts "Muhammad"
    puts "Allah"
    puts "Rakha"
  end
end
let1=Me1.new
puts let1.name_1()
#output: AAAA
puts let1.name_2()
#output: STARK
let2=Me2.new
puts let2.name_2()
#output: Muhammad Allah Rakha
```

AAAASSTARK

BOOK OF CODING

```
#POLYMORPHISM
class Me1
  attr_accessor :name
  def initialize(name)
    @name=name
  end
end
class Me2 < Me1
  def Display_1()
    "AAAA STARK"
  end
  def Display_2()
    "Muhammad Allah Rakha"
  end
end
let1=Me2.new("Yes_1")
puts let1.Display_1()
#output: AAAA STARK
puts let1.Display_2()
#output: Muhammad Allah Rakha
#MODULE
module Me
  class Me1
    attr_accessor :name
  end
  class Me2
    attr_accessor :age
  end
end
let1=Me::Me1.new
let1.name="AAAA STARK"
let2=Me::Me2.new
let2.age=20
puts let1.name
#output: AAAA STARK
puts let2.age
#output: 20
module Me
  def Me1
    return "AAAA STARK"
  end
end
class Me2
  include Me
  # here MODULE call.
end
let1=Me2.new
puts let1.Me1
#output: AAAA STARK

module Me
  def Me1()
    return "M.A.R"
  end
end
include Me
puts Me1()
#output: M.A.R
```

```
# File data reding: Muhammad Allah Rakha Naich  
File.open("rakha.txt","r").each{|anyname| puts anyname}  
#output: Muhammad Allah Rakha Naich
```

```
class Me  
  attr_reader :file  
  def initialize(filename)  
    @file=File.new(filename, "r")  
  end  
  def finished  
    @file.close  
  end  
end
```

```
let1=Me.new("rakha.txt")  
puts let1.file.gets # gets is buil-type  
#outptu: Muhammad  
puts let1.file.gets # gets is buil-type  
#output: Allah  
puts let1.file.gets # gets is buil-type  
#output: Rakha  
puts let1.file.gets # gets is buil-type  
#output: Naich  
let1.finished
```

```
# File data writing : AAAA STARK  
File.open("rakha.txt","w") do |anyname|  
  anyname.puts "AAAA STARK"  
end  
#output: AAAA STARK
```

SWIFT PROGRAMMING LANGUAGE.

MULTI-PARADIGM: PROTOCOL-ORIENTED, OBJECT-ORIENTED, FUNCTIONAL, IMPERATIVE, BLOCK STRUCTURED DECLARATIVE PROGRAMMING

Swift is a general-purpose, multi-paradigm, compiled programming language developed by Apple Inc. for iOS, iPadOS, macOS, watchOS, tvOS, and Linux. Swift is designed to work with Apple's Cocoa and Cocoa Touch frameworks and the large body of existing Objective-C code written for Apple products. It is built with the open source LLVM compiler framework and has been included in Xcode since version 6, released in 2014. On Apple platforms,[12] it uses the Objective-C runtime library which allows C, Objective-C, C++ and Swift code to run within one program

**OBJECTIVE-C,
RUST, HASKELL, RUBY, PYTHON,
C#, CLU, D**

NAME

IN THIS LANGUAGE

```
print("Muhammad Allah Rakha");
//output: Muhammad Allah Rakha
print("Muhammad Allah Rakha")
//output: Muhammad ALLah Rakha
let name1="M.A.R"
print(name1)
//output: M.A.R
var name2="AAAA STARK"
print(name2)
//output: AAAA STARK
let name3="aaaa stark"
print("My name is \(name3)")
//output: My name is aaaa stark
```

SWIFT

PROGRAMMING

```
let num1=10
let num2: Double=15
print("num1: \(num1) num2: \(num2)")
//output: num1: 10 num2: 15.0
```

```
var name="AAAA STARK"
name.insert("!",at: name.endIndex)
print(name)
//output: AAAA STARK!
```

```
var name4=["Muhammad","Allah","Rakha"]
print("\(name4[0])")
//output: Muhammad
print(name4[1])
//output: Allah
let anyone=name4[2]
print(anyone)
//output: Rakha
```

AAAASSTARK
BOOK OF CODING

BOOK OF CODING

AAAASSTARTR

```
var name5=[  
    "N1": "AAAA",  
    "N2": "STARK",  
]  
print(name5["N1"])  
//output: Optional("AAAA")  
  
var name:[String]=["M","A","R"]  
print(name)  
//output: ["M", "A", "R"]  
let n1=name.count  
for any in 0..    print(name[any])  
}  
//output: M A R  
let num1=10  
if(num1==15){  
    print("1: work")  
}else{  
    print("2: work")  
}  
//output: 2: work  
if(num1==20){  
    print("1: is work")  
}else if (num1<10){  
    print("2: is work")  
}else if (num1==10){  
    print("3: is work")  
}else{  
    print("4: is work")  
}  
//output: 3: is work  
let name=false  
let names=true  
if !name{  
    print("True Condition")  
}  
//output: True Condition  
if name && names{  
    print("Both are true")  
}else{  
    print("1: True 2: False")  
}  
//output: 1: True 2: False  
if name || names{  
    print("1: True OR 2: False")  
}else{  
    print("Both are true")  
}  
//output: 1: True OR 2: False  
  
let name1=["AAAA","STARK"]  
for anyone in name1{  
    print("\u00d7(anyone)")  
}  
//output: AAAA STARK
```

BOOK OF CODING

AAAASSTARK

```
var name="M.A.R"
for anyone in 0..<5 {
    // here(..<) is must
    // for_condition.
    print("\($anyone) : \(name)")
}
//output: 0 : M.A.R
//        1 : M.A.R
//        2 : M.A.R
//        3 : M.A.R
//        4 : M.A.R

for any in "AAAAASTARK".characters{
    print(any)
}
//output: A A A A S T A R K

let name:[Character]=["M","A","R"]
let name1=String(name)
print(name1)
//output: MAR

var name1=[  
    "Muhammad",  
    "Allah",  
    "Rakha",  
    "Naich"  
]
let count_name1=name1.count
print(count_name1)
//output: 4
for anyname in 0..<count_name1 {
    print("\($anyname+1) : \(name1[$anyname])")
}
//output: 1 : Muhammad
//        2 : Allah
//        3 : Rakha
//        4 : Naich

var num1=1
// here (var) is must
while (num1<=5){
    print(num1)
    num1=num1+1
}
//output: 1 2 3 4 5
var num2=6
repeat{
    // here (repeat) is same
    // work Like to (do)
    print(num2)
    num2=num2+1
}while(num2<=10);
//output: 6 7 8 9 10
```

AAAASSTARK

```
var name2="aaaastark"
switch name2{
    case "aaaa":
        print("1: \(name2)")
        break;
    case "stark":
        print("2: \(name2)")
        break;
    case "aaaastark":
        print("3: \(name2)")
        break;
    default:
        print("4: is now error")
}
//output: 3: aaaastark

func name1(){
    print("M.A.R")
}
name1()
//output: M.A.R
func name2()-> String{
    return "AAAA STARK"
}
print(name2())
//output: AAAA STARK
func name3()-> Int{
    return 12345
}
print(name3())
//output: 12345
func name4()-> Float{
    return 12.345
}
print(name4())
//output: 12.345
func name5()-> Double{
    return 123.4567
}
print(name5())
//output: 123.4567

func name6(anynum: Int){
    print("The number is \(anynum)")
}
name6(anynum: 12345)
//output: The number is 12345

func name7(anynum1: Int)-> Int{
    print("Number: \(anynum1)")
    //output: Number: 6789
    return anynum1
}
var name8=name7(anynum1: 6789)
print(name8)
//output: 6789
```

```
//Class
class Me1{
    func name2(name:String){
        print("Name: \(name1)")
    }
    func name3(name:String)->String{
        return "Your name is \(name1)"
    }
}
var name1="M.A.R"
var me1=Me1()
me1.name2(name: name1)
//output: Name: M.A.R
print(me1.name3(name: name1))
//output: Your name is M.A.R

class Me2{
    var name: String
    init(name: String){
        self.name=name
    }
    func name1(){
        print("The name is \(name)")
    }
}
var me2=Me2(name:"AAAA STARK")
me2.name1()
//output: The name is AAAA STARK
class Me3: Me2{
    var num: Double
    init(num: Double,name: String){
        self.num=num
        // here (super.init) is call (Me2) of
        // (init(name: String))
        super.init(name: name)
    }
    func name2(){
        print("Name is \(name)")
        print("Age is \(num)")
    }
}
var my_name="Muhammad Allah Rakha"
var me3=Me3(num: 20,name: my_name)
me3.name2()
//          Name is Muhammad Allah Rakha
//          Age is 20.0
```

```
//Enumerations
enum anyname{
    case name1(String,String,String)
    case num1 (Int)
}
var name2=anyname.name1("M","A","R")
var num2=anyname.num1(12345)
print(name2)
//output: name1("M", "A", "R")

//Structure
struct Me{
    var name: String
    var num: Int
}
var name1=Me(name: "M.A.R",num:20)
print("My name is \(name1.name)")
//output: My name is M.A.R
print("My age is \(name1.num)")
//output: My age is 20

struct Me1{
    var name="M.A.R"
    subscript(anyname: String)->String{
        return "Name: \(anyname+ " +name)"
    }
}
var me1=Me1()
print(me1["AAA STARK"])
//output: Name: AAA STARK M.A.R

struct Me2{
    var num: Int
    subscript(anynum: Int)->Int{
        print("num:\(num) and anum:\(anynum)")
        //output: num:10 and anum:20
        return num+anynum
    }
}
var me2=Me2(num: 10)
print(me2[20])
//output: 30
```

JAVASCRIPT PROGRAMMING LANGUAGE.

EVENT-
DRIVEN, FUNCTIONAL, IMPERATIVE

JavaScript (often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behavior, and all major web browsers have a dedicated JavaScript engine to execute it.

ACTIONSCRIPT, ATSCRIPT,

COFFEESCRIPT, DART,

JSCRIPT .NET, LIVESCRIPT,

OBJECTIVE-J,

OPA, QML, RAKU, TYPESCRIPT

NAME

— IN THIS LANGUAGE —



```
console.log("Muhammad Allah Rakha");
//output: Muhammad Allah Rakha
var name="AAAA STARK."
console.log(name)
//output: AAAA STARK.
console.log("My name: "+name)
//output: My name: AAAA STARK.
```

JAVA SCRIPT

— PROGRAMMING —

AAA
AA
A
BOOK OF CODING
AAA
AA
A
STARK

```
num=1234
console.log(num)
//output: 1234
num=12.445
console.log(num)
//output: 12.445
name1="AAAA"; name2="STARK";
console.log(name1+" "+name2)
//output: AAAA STARK
console.log(name1, " ", name2)
//output: AAAA STARK
var name
name="M.A.R"
console.log(name)
//output: M.A.R

var name1={
    "N1": "Muhammad",
    'N2': "Allah",
    'N3': "Rakha",
    "N4": "Naich"
}
console.log(name1["N1"])
//output: Muhammad
console.log(name1['N2'])
//output: Allah
console.log(name1['N3'])
//output: Rakha
console.log(name1["N4"])
//output: Naich
```

AAAASSSTARK

```
var name2={  
    N1: "AAAA",  
    N2: "STARK"  
}  
console.log(name2.N1)  
//output: AAAA  
console.log(name2.N2)  
//output: STARK  
console.log(name2.N1="Hassan")  
//output: Hassan  
console.log(name2.N2="Raza")  
//output: Raza  
  
name1=[1,2,3,4,5]  
console.log(name1[0])  
//output: 1  
console.log(name1[1])  
//output: 2  
name2=["Muhammad","Allah","Rakha"]  
console.log(name2[0])  
//output: Muhammad  
console.log(name2[1])  
//output: Allah  
console.log(name2[2])  
//output: Rakha  
console.log(name2.length)  
//output: 3  
  
var name3=[]  
name3.push("M")  
name3.push("A")  
name3.push("R")  
console.log(name3)  
//output: [ 'M', 'A', 'R' ]  
console.log(name3[0])  
//output: M  
console.log(name3[1])  
//output: A  
console.log(name3[2])  
//output: R  
var name4  
name4=name3.pop()  
// here (pop()) means removed  
// 1_element of array  
console.log(name3)  
//output: [ 'M', 'A' ]  
name4=name3.pop()  
// here (pop()) means removed  
// 1_element of array  
console.log(name3)  
//output: [ 'M' ]  
name4=name3.pop()  
// here (pop()) means removed  
// 1_element of array  
console.log(name3)  
//output: [] (empty)
```

BOOK OF CODING

AAAAA STARK

```
var name1=/aaaa/gi;
var name2="Wellcome Mr. aaaa STARK";
var name3=name2.replace(name1,"AAAA")
console.log(name3)
//output: Wellcome Mr. AAAA STARK
var name4=name2.split(' ',4)
console.log(name4)
//output: [ 'Wellcome', 'Mr.', 'aaaa', 'STARK' ]
console.log(name4[0])
//output: Wellcome
console.log(name4[1])
//output: Mr.
console.log(name4[2])
//output: aaaa
console.log(name4[3])
//output: STARK
var name5=name2.substr(0,8)
console.log(name5)
//output: Wellcome
var name6=name2.substring(0,13)
console.log(name6)
//output: Wellcome Mr.
console.log(name2.toUpperCase())
//output: WELLCOME MR. AAAA STARK
console.log(name2.toLocaleLowerCase())
//output: wellcome mr. aaaa stark

var name="aaaa"
switch(name){
    case "aaaastark":
        console.log("1: work")
        break;
    case "aaaa":
        console.log("2: work")
        break;
    case "stark":
        console.log("3: work")
    default:
        console.log("Error now")
}
//output: 2: work

var num1=0
while(num1<=5){
    console.log(num1)
    num1++;
}
//output: 0 1 2 3 4 5
```

AAAAA STARK

BOOK OF CODING

```
num2=6
do{
    console.log(num2)
    num2++;
}while(num2<=10);
//output: 6 7 8 9 10

var num3;
for(num3=0; num3<=4;num3++){
    console.log("M.A.R")
}
//output: M.A.R
//          M.A.R
//          M.A.R
//          M.A.R
//          M.A.R

var name1=["Muhammad","Allah","Rakha"]
for(anyone in name1){
    console.log(name1[anyone]+ " ")
}
//output: Muhammad Allah Rakha

var num1=0
while(num1<=5){
    console.log(num1)
    if(num1==3){
        break;
    }
    num1++;
}
//output: 0 1 2 3

//Function
Name1=function(){
    console.log("M.A.R")
}
Name1();
//output: M.A.R

Name2=function () {
    return "AAAA STARK"
}
name1=Name2()
console.log(name1)
//output: AAAA STARK
```

AAAASSТАRK

BOOK OF CODING

```
var name2,name3;
name2="Hassan";
name3="Raza";
Name2=function (n2,n3) {
    console.log("name2: "+n2)
    //output: name2: Hassan
    console.log("name3: "+n3)
    //output: name4: Raza
    return n2+n3
    // name2+name3
}
name4=Name2(name2,name3)
console.log("name4: ",name4)
//output: HassanRaza
```

```
var name5,name6
name5={
    "N1": "Muhammad",
    'N2': "Allah",
    N3: "Rakha"
};
name6= function (name_5){
    console.log(name_5["N1"])
    //output: Muhammad
    console.log(name_5['N2'])
    //output: Allah
    console.log(name_5.N3)
    //output: Rakha
}
name6(name5);
```

AAAASSTAAARK

BOOK OF CODING

```
var name1=function(n1,n2,n3){  
    var name2={}  
    name2.n1=n1  
    name2.n2=n2  
    name2.n3=n3  
    return name2  
}  
  
var name3  
name3=name1("M","A","R")  
console.log(name3.n1)  
//output: M  
console.log(name3.n2)  
//output: A  
console.log(name3.n3)  
//output: R  
  
  
var name1=function (n1,n2,n3) {  
    this.n1=n1  
    this.n2=n2  
    this.n3=n3  
  
    this.name2=function(){  
        var name3  
        name3=this.n1+" "  
        name3+=this.n2+" "  
        name3+=this.n3  
        console.log(name3)  
    };  
};  
  
var names =new name1("M","A","R")  
names.name2()  
//output: M A R
```

PERL PROGRAMMING LANGUAGE.

MULTI-
**PARADIGM: FUNCTIONAL, IMPERATIVE,
OBJECT-ORIENTED (CLASS-
BASED), REFLECTIVE**

Perl is a family of two high-level, general-purpose, interpreted, dynamic programming languages. "Perl" refers to Perl 5, but from 2000 to 2019 it also referred to its redesigned "sister language", Perl 6, before the latter's name was officially changed to Raku in October 2019

**INFLUENCED
BY AWK, BASIC, C, C++, LISP,
SED, UNIX SHELL**

NAME

IN THIS LANGUAGE

```
print("Muhammad Allah Rakha");
#output: Muhammad Allah Rakha
$names="M.A.R";
print "$names and $names";
#output: M.A.R and M.A.R
print "${names} and ${names}";
#output: M.A.R and M.A.R
print("Enter your Name. ");
# Enter your Name. AAAA STARK
$name=<STDIN>;
print("Your name is $name");
#output: Your name is AAAA STARK
print("Enter your Age. ");
# Enter your Age. 20
$num=<STDIN>;
print("Your age is $num");
#output: Your age is 20
```

PERL

PROGRAMMING

```
$name1=12;
print "name1 is $name1";
#output: name1 is 12
$name2=12.33;
print $name2;
#output: 12.33
print $name2+22.4444;
# here use (/,+,-,*,^,%)
#output: 34.7744
$n1=100;
$n1++;
# here use (--,++, **, *, /, %)
print $n1;
#output: 101
```

AAAASSTTAAKK
BOOK OF CODING

```
@name1=("Muhammad","Allah","Rakha");  
# declear a Array mathod use (@)  
# and for print array use ($)  
print $name1[0]," ",$name1[1]," ",$name1[2];  
#output: uhammad Allah Rakha  
print @name1;  
#output: MuhammadAllahRakha  
print @name1[0..2];  
#output: MuhammadAllahRakha  
print $#name1; # for length  
#output: 2  
push (@name1,"Naich");  
print @name1;  
#output: MuhammadAllahRakhaNaich  
for ($i=0;$i<$#name1; $i++){  
    print "$name1[$i] ";  
}  
#output: Muhammad Allah Rakha  
for ($i=0;$i<=5;$i++){  
    print("$i ");  
}  
#output: 0 1 2 3 4 5  
foreach $i (@name1){  
    print ("$i ");  
}  
#output: Muhammad Allah Rakha Naich  
foreach $o (@name1){  
    $o="M.A.R";  
}  
print @name1;  
#output: M.A.RM.A.RM.A.RM.A.R
```

```
@name1=("Muhammad","Allah","Rakha");  
@reversed_array=reverse(@name1);  
# reverse find  
print @reversed_array;  
#output: RakhaAllahMuhammad  
@sort_array=sort(@name1);  
print @sort_array;  
#output: AllahMuhammadRakha
```

```
@name2=(10,99,7,47,0);  
@sort_name2=sort{$a <=> $b} (@name2);  
print "@sort_name2 ";  
#output: 0 7 10 47 99  
@sort_name2=sort{$b <=> $a} (@name2);  
print "@sort_name2 ";  
#output: 99 47 10 7 0
```

```
@name3=("M","A","R");  
@name4=qw(M A R);  
print "@name3 ","@name4";  
#output: M A R M A R
```

```
$a="aaaa";  
$b="stark";  
@name5=qw($a $b);  
print "$name5[0] and $name5[1]";  
#output: $a and $b
```

```
$name1= ("n1"=>"Muhammad",
        "n2"=>"Allah",
        "n3"=>"Rakha");
```

```
# we use (%) for create
# associative arrays
```

```
print "$name1{n1} ";
print "$name1{n2} ";
print "$name1{n3} ";
#output: Muhammad Allah Rakha
```

```
$num1=12;
$num2=10;
if($num1<$num2){
    print("1: is work");
}else{
    print("1: is not work");
}
#output: 1: is work
```

```
if($num1<$num2){
    print("1: is work");
}elseif ($num1>$num2){
    print("2: is work");
}
#output: 2: is work
```

```
$name1="aaaastark";
if(defined($name1 != "aaaastark")){
    # defined case in which (false)
    # here me use (!=). if(statement) is false.
    print ("Yes is equal");
}else {
    print ("Not is equal");
}
#output: Yes is equal
```

```
$num=1;  
while ($num<=4) {  
    print("$num ");  
    $num++;  
}  
#output: 1 2 3 4
```

```
do{  
    print("$num ");  
    $num++;  
}while ($num<=5);  
#output: 1 2 3 4 5
```

```
$num2=10;  
if(($num2>100) && ($num2<5)){  
    print("Condition is correct");  
}else {  
    print("Condition is wrong");  
}  
#output: Condition is wrong
```

```
if(($num2>8) || ($num2<5)){  
    print("Condition is correct");  
}else {  
    print("Condition is wrong");  
}  
#output: Condition is correct
```

```
if(!$num2==10)){  
    print("Not condition is not work");  
}else {  
    print("Not condition is work");  
}  
#output: Not condition is work
```

AAAASSTAAARK

BOOK OF CODING

```
$name1="Muhammad>Allah:Rakha";
@name2=split(/:/,$name1);
for $i (@name2){
    print $o++," $i\n";
    # $o++ (me defined to 0 1 2)
}
```

```
#output: 0 Muhammad
#       1 Allah
#       2 Rakha
```

```
$name3=join(".",@name2);
print ("$name3 ");
#output: Muhammad.Allah.Rakha
```

```
$name4="aaaa stark";
print (substr($name4,5,9));
#output: stark
```

```
print (index($name4,"stark"));
#output: 5
```

```
$num1=rand(12);
print("$num1");
#output: 1.0579833984375
```

FUNCTIONS

```
# sub (is define a Function)
# call a function by (&)
```

```
sub name1{
    print("Muhammad Allah Rakha");
}
&name1;
#output: Muhammad Allah Rakha
```

```
sub name2{
    print("Name: $a.$b.$c");
}
$a="M"; $b="A"; $c="R";
&name2;
#output: Name: M.A.R
```

```
sub name3{
    print("1: $_[0]");
    print("2: $_[1]");
    # here $_ is must
    # me use two name.
    # so, $_[0] = aaaa
    # so, $_[1] = stark
}
$names1="aaaa";
$names2="stark";
&name3($names1,$names2);
#output: 1: aaaa 2: stark
```

LUA PROGRAMMING LANGUAGE.

MULTI-PARADIGM: SCRIPTING, IMPERATIVE (PROCEDURAL, PROTOTYPE-BASED, OBJECT-ORIENTED), FUNCTIONAL

Lua (lua meaning moon) is a lightweight, high-level, multi-paradigm programming language designed primarily for embedded use in applications. Lua is cross-platform, since the interpreter of compiled bytecode is written in ANSI C and Lua has a relatively simple C API to embed it into applications

**GAMEMONKEY, IO, JAVASCRIPT,
JULIA, MINID, RED,
RING, RUBY, SQUIRREL,
MOONSCRIPT, C-- ,
C++, CLU, MODULA, SCHEME,
SNOBOL**

NAME

— IN THIS LANGUAGE

AAAAA STARK
BOOK OF CODING

```
print("Muhammad Allah Rakha")
--output: Muhammad ALLah Rakha
print("AAAA STARK")
--output: AAAA STARK
io.write("M.A.R")
--output: M.A.R
name1={n1=print}
name1.n1("AAAA STARK")
--output: AAAA STARK
local name2,name3="AAAA","STARK"
print("name2 and name3 ",name2..name3)
--output: name2 and name3  AAAASTARK
print("length of name2 ",#name2)
--output: Length of name2  4
print("length of name3 ",#name3)
--output: Length of name3  5
print("Enter your Name.")
name4=io.read()
--output: Enter your Name.
--          AAAA STARK
print("Your Name: ",name4)
--output: Your Name: AAAA STARK
```

LUA

PROGRAMMING

AAAAA STARK
BOOK OF CODING

```
local name="M.A.R"  
io.write(name)  
--output: M.A.R
```

```
name1="AAAA STARK"  
io.write(name1)  
--output: AAAA STARK
```

```
print(type(name1))  
--output: string
```

```
local n1=12345  
print(n1)  
--output: 12345
```

```
local a,b=5,10  
print(a," ",b)  
--output: 5 10
```

```
a,b=10.11, 55.22  
print(a," ",b)  
--output: 10.11 55.22
```

```
n2=b/a  
print("n2 =",n2)  
--output: n2 = 5.461918892186
```

```
n3=b+a  
print("n3 =",n3)  
--output: n3 = 65.33
```

```
n4=b*a  
print("n4 =",n4)  
--output: n4 = 558.2742
```

```
local n5,n6=10,20
if(a==b)
then
    print("n5 and n6 is equal")
else
    print("n5 and n6 not_equal")
end
--output: n5 and n6 not_equal
```

```
if((n5>n6) and (n5==n6))
then
    print("Condition is true")
else
    print("Condition is not_true")
end
--output: Condition is not_true
```

```
if((n5<n6) or (n5==n6))
then
    print("Condition is true")
else
    print("Condition is not_true")
end
--output: Condition is true
```

```
if((n5<n6) and (not(n5==n6)))
then
    print("Condition is true")
else
    print("Condition is not_true")
end
--output: Condition is true
```



AAAASSTARK

BOOK OF CODING

```
name1="aaaa STARK"
name2=string.gsub(name1,"aaaa","AAAA")
print(name2)
--output: AAAA STARK

name3="aaaa.."stark"
print(name3)
--output: aaaastark

name4=tonumber("10e4")
print(name4)
--output: 100000

name5=string.reverse(name3)
print(name5)
--output: kratsaaaa

name6=string.rep("M.A.R",4)
print(name6)
--output: M.A.RM.A.RM.A.RM.A.R

name7=string.lower(name2)
print(name7)
--output: aaaa stark

name8=string.upper(name7)
print(name8)
--output: AAAA STARK

name9=string.sub(name8,1,-1)
print(name9)
--output: AAAA STARK

name10=string.format("x =%d and y =%d",10,20)
print(name10)
--output: x =10 and y =20

name11=string.find(name9,"AAAA")
print(name11)
--output: 1 (true)

name12=insert("aaaa .....",6,"stark")
print(name12)
--output: aaaa stark
```

AAAASSТАРК

BOOK OF CODING

```
name1={}
name1[1]="AAAA"
name1[2.1]='x'
name1["A"]=12.44
name1['a']=11
```

```
print(name1[1])
--output: AAAA
```

```
print(name1[2.1])
--output: x
```

```
print(name1["A"])
--output: 12.44
```

```
print(name1['a'])
--output: 11
```

```
name2={"Muhammad","Allah","Rakha"}
for i=1,#name2 do print(name2[i]) end
--output: Muhammad Allah Rakha
```

```
name1={}
for i=1,5 do
    name1[i]=io.read()
--Open Terminal and run file
--put: 1 2 3 4 5
--Here you can put(string,float,etc)
end
```

```
for i=1,#name1 do
    print(name1[i])
end
--output: 1 2 3 4 5
```

```
local name1=10
local name2=15
if name1==name2 then
    print("Both are not_equal")
elseif (not(name1==name2))then
    print("Both are equal")
else
    print("Bot are same")
end
--output: Both are equal

--Functions
function me1()
-- function is first_
-- character also small
io.write(
    "M.A.R")
end
me1()
--output: M.A.R

function me2(name1,name2)
    print("name1 =",name1)
    print("name2 =",name2)
end
me2("aaaa","stark")
--output: name1 =    aaaa
--           name2 =    stark

function me3()
    for i=1,5 do
        print(i," + ",i,"=",i+i)
    end
end
me3()
--output: 1 + 1 = 2
--         2 + 2 = 4
--         3 + 3 = 6
--         4 + 4 = 8
--         5 + 5 = 10
```

```
function me4(num)
    return num
end
print(me4(5))
--output: 5

function me5(...)
    print("Anyone ",...)
    name3=select(...)
    return name3
end
me5(123)
--output: Anyone 123

print(me5(123))
--output: Anyone 123_
--          null_"

local name1=10
local name2=5
while name2<name1 do
    print("name2 =",name2)
    name2=name2+1
end
--output: name2 =      5
--          name2 =      6
--          name2 =      7
--          name2 =      8
--          name2 =      9

while name2<=name1 do
    print("name2 =",name2)
    name2=name2+1
end
--output: name2 =      5
--          name2 =      6
--          name2 =      7
--          name2 =      8
--          name2 =      9
--          name2 =     10
```

PHP PROGRAMMING LANGUAGE.

**IMPERATIVE, FUNCTIONAL, OBJECT-
ORIENTED, PROCEDURAL, REFLECTIVE**

PHP is a general-purpose scripting language that is especially suited to web development. It was originally created by Danish-

Canadian programmer Rasmus Lerdorf in 1994; the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

**PERL, HTML, C, C++, JAVA, TCL
JAVASCRIPT, HACK**

NAME

— IN THIS LANGUAGE

AAAAA STARK
BOOK OF CODING

```
<?php
```

```
printf("Muhammad Allah Rakha");  
//output: Muhammad ALLah Rakha
```

```
echo ("M.A.R");  
//output: M.A.R
```

```
printf("%s ", "AAAA STARK");  
//output: AAAA STARK
```

```
printf("AAAA STARK");  
//output: AAAA STARK
```

```
printf("\n");  
//output: endl (next line)
```

```
printf("M.A.R");  
//output: M.A.R
```

```
echo ("Muhammad Allah Rakha");  
//output: Muhammad ALLah Rakha
```

```
?>
```

PHP

— PROGRAMMING —

BOOK OF CODING
AAAASSTARK

```
<?php
$name1=123;
echo $name1;
//output: 123
$name2=12.3456;
echo $name2;
//output: 12.3456
$name3=$name1+$name2;
echo $name3;
//output: 135.3456
$name4=null;
echo $name4;
//output: null
$name5=true;
echo $name5;
//output: ture (1)
$name5=false;
echo $name5;
//output: false (0)
$name6=array(1,2,3,4,5);
echo $name6[0];
//output: 1
echo $name6;
//output: Array
$name7=[ "aaaa", "stark", 123=>"M.A.R"];
printf($name7[0]); echo ($name7[1]);
//output: aaaa      stark
echo $name7[123]; //output: M.A.R
$name8=new stdClass(); //Objects
$name8->n1="AAAA STARK";
echo $name8->n1;
//output: AAAA STARK
$name9=(object)[ "M"=>"M.A.R"];
echo $name9->M; #here (1, "1") not use
//output: M.A.R
?>
```



AAAASSTAAARK

BOOK OF CODING

```
<?php  
const PI=3.1416;  
echo PI;  
//output: 3.1416  
const Name="M.A.R";  
echo Name;  
//output: M.A.R  
var_dump(1==1);  
//output: bool(true)  
$name2=(int)123.55;  
print $name2;  
//output: 123  
$name3=(string)12456;  
print $name3;  
//output: 12456(string)  
# here you use (float,boolval)  
$name1=10;  
if($name1==10){  
    print("Yes equal");  
}  
//output: Yes equal  
if($name1!=10){  
    print("No equal");  
}  
else{  
    print("Yes equal now.");  
}  
//output: Yes equal now.  
$name4=<<<EOF  
My name is Muhammad Allah Rakha.  
I have a study in University.  
Me from Lahore.  
EOF;  
var_dump($name4);  
//output: string(80) "My name is Muhammad Allah Rakha.  
// I have a study in University.  
// Me from Lahore."  
echo $name4;  
//output: My name is Muhammad Allah Rakha.  
// I have a study in University.  
// Me from Lahore.  
?>
```

```
<?php
print(3 <= 2);
//output: 1
$num1=1;
echo ++$num1;
//output: 1
echo $num1++;
//output: 2
echo $num1--;
//output: 3
echo --$num1;
//output: 1
$num2=10;
$num3=15;
$name1=($num2==15)&&($num3==15);
echo $name1;
//output: 0
var_dump($name1);
//output: bool(false)
$name2=($num2==10) || ($num3>15);
echo $name1;
//output: 1
var_dump($name2);
//output: bool(true)
$name3="M.A.R";
$name4=&$name3;
echo $name4;
//output: M.A.R
?>
```

BOOK OF CODING

AAAASSТАРК

```
<?php  
$name1=array("Muhammad","Allah","Rakha");  
echo $name1[0];  
echo $name1[1];  
echo $name1[2];  
//output: MuhammadAllahRakha  
$name2=[ "M", "A", "R" ];  
echo $name2[0];  
echo $name2[1];  
echo $name2[2];  
//output: MAR  
$name3=array(  
    "M1"=>"M",  
    "M2"=>"A",  
    "M3"=>"R"  
);  
echo $name3["M1"];  
echo $name3["M2"];  
echo $name3["M3"];  
//output: MAR  
$name4=[  
    "M1"=>"M",  
    "M2"=>"A",  
    "M3"=>"R"  
];  
echo $name4["M1"];  
echo $name4["M2"];  
echo $name4["M3"];  
//output: MAR  
$name5=array();  
$size=4;  
$name6="M.A.R";  
for($i=0;$i<$size;$i++){  
    $name5[$i]=$name6;  
}  
for($i=0;$i<$size;$i++){  
    print ("$name5[$i] ");  
}  
//output: M.A.R M.A.R M.A.R M.A.R  
print(is_array($name5));  
//output: true (1)  
$name7=[ "M", "A", "R" ];  
foreach ($name7 as $i){  
    print("$i ");  
}  
//output: M A R  
?>
```



BOOK OF CODING

AAAASSTARK

```
<?php  
function name(){  
    print("M.A.R");  
}  
name();  
//output: M.A.R  
function name1($names,$num){  
    print("$names and $num");  
}  
function name3($name){  
    return ($name);  
}  
$retrun_name3=name3("M.A.R");  
echo $retrun_name3;  
//output: M.A.R  
name1("AAAA STARK",123);  
//output: AAAA STARK 123  
function name2($name){  
    switch ($name){  
        case "aaaa":  
            print("1: $name");  
            break;  
        case "stark":  
            echo ("2: $name");  
            break;  
        case "aaaastark":  
            print("3: $name");  
            break;  
        default:  
            print("Condition is not work");  
            break;  
    }  
}  
name2("aaaastark");  
//output: 3: aaaastark  
name2("aaaa");  
//output: 1: aaaa  
name2("stark");  
//output: 2: stark  
name2(124455);  
//output: Condition is not work  
?>
```

R PROGRAMMING LANGUAGE.

MULTI-PARADIGM: ARRAY, OBJECT-ORIENTED, IMPERATIVE, FUNCTIONAL, PROCEDURAL, REFLECTIVE

R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis.

Polls, data mining surveys, and studies of scholarly literature databases show substantial increases in popularity; as of September 2020, R ranks 9th in the TIOBE index, a measure of popularity of programming languages.

COMMON
LISPSSCHEME, XLISPSTAT

AAAAA STARK



NAME

— IN THIS LANGUAGE —

```
print("AAAA STARK")
#output: AAAA STARK
```

```
name1="Muhammad Allah Rakha"
print(name1)
#output: Muhammad ALLah Rakha
```

```
name2 <- "M.A.R"
print(name2)
#output: M.A.R
```

```
name9 <- 'Muhammad Allah Rakha'
print(name9)
#output: Muhammad ALLah Rakha
```



#DATA TYPES

Vectors

```
name3 <- c("Muhammad", "Allah", "Rakha")
print(name3)
#output: "Muhammad" "Allah" "Rakha"
```

Lists

```
name4 <- list(c("M.A.R"), 44.3, 22, "M.A.R")
print(name4)
#output: "M.A.R" 44.3 22 "M.A.R"
```

```
# Matrices
name5=matrix(c("Muhammad","Allah","Rakha","Naich"),
             ,nrow=2,ncol=2)
print(name5)
#output: "Muhammad" "Rakha"
#           "Allah"    "Naich"

# Factors
name6 <- c("Muhammad","Allah","Rakha","Naich")
name7 <-(factor(name6))
print(name7)
#output: Muhammad Allah Rakha Naich
#       Levels: Allah Muhammad Naich Rakha

# Data Frames
name8 <- data.frame(
  gender = c("Male"),
  name = c("M.A.R"),
  age = c(20)
)
print(name8)
#output:   gender  name age
#       1   Male M.A.R  20

# Formats
name1 <- format(22.33344, digits=4)
print(name1)
#output: 22.33
name2 <- format(c(123,567.89),scientific= TRUE)
print(name2)
#output: 1.2300e+02" "5.6789e+02
name3 <- format(123.456,nsmall=5)
print(name3)
#output: 123.45600
```

AAAASSTARK

BOOK OF CODING

```
#To upper and To Lower
name4 <- toupper("aaaa stark")
print(name4)
#output: AAAA STARK
name5 <- tolower("AAAA STARK")
print(name5)
#output: aaaa stark

#Substring case
name6 <- substring("aaaastark",5,9)
print(name6)
#output: stark

# Add,Sub,Mul,Div
name1 <- c(2,3,4,5)
name2 <- c(6,7,8,9)
name3 <- name1+name2
print(name3)
#output: 8 10 12 14
name4 <- name1*name2
print(name4)
#output: 12 21 32 45
name5 <- name2-name1
print(name5)
#output: 4 4 4 4
name6 <- name2/name1
print(name6)
#output: 3.000000 2.333333
#           2.000000 1.800000

name1 = c(5, FALSE, 2)
name2 <- c(2, TRUE, 6)
print.default(name1+name2)
#output: 7 1 8
print.default(name1>name2)
#output: TRUE FALSE FALSE
print.default(name1&name2)
#output: TRUE FALSE TRUE
print.default(name1|name2)
#output: TRUE TRUE TRUE
```

BOOK OF CODING



```
# DECISION MAKING
num1 <- 10
if(num1>5){
  print("Grater")
}else{
  print("Lower")
}
#output: Grater
num2 = 22
if(num2 >100){
  print("Big")
}else if(num2>20){
  print("Small")
}else{
  print("Medium")
}
#output: Small
#LOOPS
name1 <- c("Muhammad","Allah","Rakha","Naich")
num3 <- 1
while(num3<=4){
  print(name1)
  num3=num3+1
}
#output: "Muhammad" "Allah"      "Rakha"      "Naich"
#           "Muhammad" "Allah"      "Rakha"      "Naich"
#           "Muhammad" "Allah"      "Rakha"      "Naich"
#           "Muhammad" "Allah"      "Rakha"      "Naich"
for(i in name1){
  print(i)
}
#output: "Muhammad" "Allah" "Rakha" "Naich"

name2 <- c("Muhammad","Allah","Rakha","Naich")
for(i in name2){
  if(i == "Rakha"){
    next
    # next concept
  }
  print(i)
}
#output: Muhammad Allah Naich
for(i in name2){
  if(i == "Rakha"){
    break
    # break concept
  }
  print(i)
}
#output: Muhammad Allah
```

```
# FUNCTIONS MAKING

Name1 <- function(){
  print("Muhammad")
  print("Allah")
  print("Rakha")
}
Name1()
#output: Muhammad Allah Rakha

Name2 <- function(name1,num1){
  print(name1)
  print(num1)
}
Name2("M.A.R",12345)
#output: M.A.R 12345
Name2(name1 = "AAAA",6789)
#output: AAAA 6789

Name3 <- function(name2){
  for(i in name2){
    print(i+i)
  }
}
Name3(4)
#output: 8

Me1 <- function(name,namess){
  print(sprintf("One: %s and %s",name,namess))
  cat("Two: ",name," and ",namess)
  sprintf("Three: %s and %s",name,namess)
}
Me1("AAAA STARK","M.A.R")
#output: One: AAAA STARK and M.A.R
#output: Two: AAAA STARK and M.A.R
#output: Three: AAAA STARK and M.A.R
```

#SEQUENCES, REPETITION, SORTING

```
name1 <- seq(from=12,to=44,by=3)
print(name1)
# from+by == to (12+3)=15 upto (44-2)=42
#output: 12 15 18 21 24 27 30 33 36 39 42
```

```
name2 <- rep(x=1,times=5)
```

```
print(name2)
#output: 1 1 1 1 1
```

```
name3 <- rep(x=c(1,2,3),times=3)
```

```
print(name3)
#output: 1 2 3 1 2 3 1 2 3
```

```
name4 <- sort(x=c(2.1,4.2,5.22,1.2),
```

```
| decreasing = FALSE)
```

```
print(name4)
#output: 1.20 2.10 4.20 5.22
```

```
name5 <- sort(x=c(2.1,4.2,5.22,1.2),
```

```
| decreasing = TRUE)
```

```
print(name5)
#output: 5.22 4.20 2.10 1.20
```

```
# DATA ENCRYPTED
```

```
name1 <- "AAAA STARK"
```

```
save(name1,file = "Rakha.rdata")
```

```
# save a data in encrypted form
```

```
# of file Rakha.rdata
```

```
rm(name1)
```

```
# remove (name1).Here can't
```

```
# print the name1.
```

```
load("Rakha.rdata")
```

```
# here Load(Rakha.rdata) file.
```

```
print.default(name1)
```

```
#output: AAAA STARK
```

RUST PROGRAMMING LANGUAGE.

MULTI-PARADIGM: CONCURRENT, FUNCTIONAL, GENERIC, IMPERATIVE, STRUCTURED

Rust is a multi-paradigm programming language focused on performance and safety, especially safe concurrency. Rust is syntactically similar to C++, and provides memory safety without using garbage collection

ALEF, C#, C++, CYCLONE,
ERLANG, HASKELL, LIMBO,
NEWSQUEAK, OCAML, RUBY,
SCHEME, STANDARD ML, SWIFT

NAME

— IN THIS LANGUAGE

```
// fn main() is required
// and print! and ;
// is must used in RUST
fn main(){
    print!("AAAA STARK");
    //output: AAAA STARK

    print!("{} {} {}", "Muhammad", "Allah", "Rakha");
    //output: Muhammad Allah Rakha

    println!("Muhammad Allah Rakha");
    //output: Muhammad Allah Rakha

    print!("\n");
    //output: endl (next lin)

    println!("Hassan Raza");
    //output: Hassan Raza
}
```

RUST

— PROGRAMMING

```
fn main(){
    let name="M.A.R";
    println!(" {} ",name);
    //output: M.A.R
}
```

AAAASSTARK
BOOK OF CODING

BOOK OF CODING

AAAASSTAAARK

```
fn main(){
    let num1=10;
    let num2=20;
    print!("The add {}+{}={}",num1,num2,num1+num2);
    //output: The add 10+20=30

    // Mutable Variables
    let mut num3=33;
    print!("num3 is {}",num3);
    //output: num3 is 33
    // mut is to over_write value
    num3=66;
    print!("num3 is {}",num3);
    //output: num3 is 66

    let num4;
    let num5=10;
    num4=num5;
    print!("num4 is {}",num4);
    //output: num4 is 10

    // Leading Underscore
    let number=10;
    let _number=15;
    println!("number={}",number);
    //output: number=10
    println!("_number={}",_number);
    //output: _number=15

    let name1=true;
    let name2=false;
    print!("name1={}",name1);
    //output: true
    print!("name2={}",name2);
    //output: false
    print!("{}",name1||name2 && !name1);
    //output: true
    print!("{}",(true || true)&& !name1);
    //output: false
}
```

BOOK OF CODING

AAASSTARK

```
// CONTROLLING EXECUTION FLOW

fn main(){
    let num1=10;
    if num1>5 {print! ("num1={}",num1);}
    //output: num1=10
    let num2=20;
    if num2>20 {print!("if_num2={}",num2);}
    else{print!("else_num2={},num2");}
    //output: else_num2=20
    let num3=30;
    // 1_way
    if num3>100{print!("Big num3={},num3");}
    else if num3>0 {print!("Small num3={},num3");}
    else {print!("Negitive num3={},num3");}
    //output: Small num3=30
    // 2_way
    print!("{}","if num3>100 {"Big"}
            else if num3>0 {"Small"}
            else {"Negitive"});
    //output: Small
}

fn main(){
    //Conditional Loops
    let mut num4=1;
    while num4 <=5{
        print!("{} ", num4);
        num4+=1;
    }
    //output: 1 2 3 4 5
    let mut num5=1;
    while num5 <=50{
        num5+=1;
        if num5%3 == 0 {continue;}
        if num5 * num5 >500 {break;}
        print!("{}\t",num5 * num5);
    }
    //output: 123454 16 25 49 64
    //          100 121 169 196 256
    //          289 361 400 484

    // Counting Loops
    for i in 1..=10 {
        print!("{} ",i * i);
    }
    //output: 1 4 9 16 25 36
    //          49 64 81 100
    let mut num6=5;
    for y in 1..=num6{
        num6-=1;
        print!("{} ",y);
    }
    //output: 1 2 3 4 5
}
```

BOOK OF CODING



// DATA SEQUENCES (ARRAY)

```
fn main(){
    let n = ["Muhammad", "Allah", "Rakha"];
    print!("{} {} {}", n[0], n[1], n[2]);
    //output: Muhammad Allah Rakha
    print!("length of n={}", n.len());
    //output: Length of n=3
    let mut name1=["x", "y"];
    print!("{} {}", name1[0], name1[1]);
    //output: x y
    name1=["1", "2"];
    print!("{} {}", name1[0], name1[1]);
    //output: 1 2
    let mut name2=[1; 5];
    for i in 0..name2.len(){
        name2[i]+=1;
        // 5-times is run of ([ 5])
        // then add 1 by 1 in ([1;])
    }
    print!("Array of name2 ={:?}", name2);
    //output: Array of name2 =[2, 2, 2, 2, 2]
    for i in 0..name2.len(){
        print!("{} ", name2[i]);
    }
    //output: 2 2 2 2 2
}
```

//POINTER REFERENCES

```
fn main(){
    let num1: i128=10;
    let num2: i128=15;
    let num3: &i128=&num1;
    let num4: &i128=&num2;
    print!("{} ", *num3);
    //output: 10
    print!("{} ", *num4);
    //output: 15
    let mut num6: i128=20;
    let mut num7: i128=25;
    let mut num8: &mut i128=&mut num6;
    let mut num9: &mut i128=&mut num7;
    print!("{} ", *num8);
    //output: 20
    *num8+=1;
    print!("{} ", *num8);
    //output: 21
    print!("{} ", *num9);
    //output: 25
    *num9+=2;
    print!("{} ", *num9);
    //output: 27
}
```

```
//DEFINING FUNCTIONS
// Function object name is Small Characters.
fn me1(){
    print!("{} ","Muhammad");
    print!("{} ","Allah");
    print!("{} ","Rakha");
}
fn me3(num: i128,name1: &str){
// str is not useable
// but &str is useable for string
    print!("Num={}",num);
    print!("Name={}",name1);
}
fn me4(mut num1: f64,num2: f64){
    num1=22.3;
    print!("Num1={} is change. ",num1);
    print!("Num2={}",num2)
}
fn main(){
    me1();
    //output:Muhammad Allah Rakha
    fn me2(){print!("{}","M.A.R");}
    me2();
    //output: M.A.R
    me3(12345,"AAA STARK");
    //output: Num=12345 Name=AAA STARK
    me4(22.2,44.4);
    //output: Num1=22.3 is change. Num2=44.4
}
```

GO PROGRAMMING LANGUAGE.

MULTI-
PARADIGM: CONCURRENT, FUNCTIONAL,
IMPERATIVE, OBJECT-ORIENTED

Go is a statically typed, compiled programming language designed at Google by Robert Griesemer, Rob Pike, and Ken Thompson. Go is syntactically similar to C, but with memory safety, garbage collection, structural typing, and CSP-style concurrency. The language is often referred to as "Golang" because of its domain name, golang.org, but the proper name is Go. There are two major implementations: Google's self-hosting compiler toolchain targeting multiple operating systems, mobile devices, and WebAssembly.gccgo, a GCC frontend.

ALEF, APL, BCPL,
C, CSP, LIMBO, MODULA, NEWS
QUEAK, OBERON, OCCAM, PASC
AL, PYTHON, SMALLTALK

NAME

— IN THIS LANGUAGE

AAAAA STARK
BOOK OF CODING

```
package main

import "fmt"

func main(){
    fmt.Println("AAAA STARK")
    fmt.Println(`Muhammad
                Allah
                Rakha`)
    //output: Muhammad
    //          Allah
    //          Rakha
}
```

GO

PROGRAMMING

```
package main

import "fmt"

func main(){
    var name1=888
    name2:=22.44;
    fmt.Println("Name1=",name1)
    //output: Name1=888
    fmt.Println("Name2=",name2)
    //output: Name2=22.44
    var n1,n2,n3="Muhammad","Rakha","Naich"
    fmt.Println(n1," ",n2," ",n3)
    //output: Muhammad Allah Rakha
    n4:="AAAA"; n5:="STARK"
    fmt.Println(n4," ",n5)
    //output: AAAA STARK
    const name3 int64=10
    const name4 float64=30.33
    fmt.Println(name3," ",name4)
    //output: 10 30.33
}
```

```
package main

import "fmt"

func main(){
    var n6 int64 =33
    var n7 int64 =22
    var n8 int64; n8=n7+n6
    fmt.Println("n7+n6=",n8)
    //output: n7+n6=55
    n8=n6-n7; fmt.Println("n6-n7=",n8)
    //output: n6-n7=11
    var name5="aaaastark"
    if name5 == "AAAA" {
        fmt.Println("1: ",name5)
    }else if name5 == "aaaastark"{
        fmt.Println("2: ",name5)
    }else if name5 == "STARK"{
        fmt.Println("3: ",name5)
    }
    //output: 2: aaaastark
}

package main

import "fmt"

func main(){
    var n1=1200
    var n2=n1%400 == 0 || (n1%4 == 0 && n1%100 != 0)
    fmt.Println("The Leap Year is now ",n2)
    //output: true

    var name1 ="rakha"
    fmt.Println("the swich command use now.")
    switch name1{
    case "allah rakha":
        fmt.Println("1: ",name1)
    case "rakha","RAKHA":
        fmt.Println("2: ",name1)
    case "rakha naich","Rakha":
        fmt.Println("3: ",name1)
    default:
        fmt.Println("Variable is not work now")
    }
    //output: the swich command use now. 2: rakha
}
```

```
package main

import "fmt"

func main(){
    var name2="aaaastark"
    switch{
        case name2 != "aaaa":
            fmt.Println("AAAA STARK")
        case name2 == "stark":
            fmt.Println("aaaa stark")
        default:
            fmt.Println("Not work variable")
    }
    //output: AAAA STARK
    //          DAY AND MONTH AND YEAR
    name1 := 43224.99345
    fmt.Println("Folat = '4.4f' = %4.4f =", name1)
    //output: Folat = '4.4f' = %4.4f =43224.99345
    name2:=0.0
    for i:=0;i<11;i++{
        name2+=0.1
    }
    fmt.Println("Name2= %v",name2)
    //output: Name2= %v1.0999999999999999
    fmt.Println("Name2= %T",name2)
    //output: Name2= %T1.0999999999999999
}

package main
import "fmt"
func kelcintоКельвины(K float64) float64{
    K-=273.15
    return K
}
func main() {
    kelvin:= 294.0
    celsius:= kelcintоКельвины(kelvin)
    fmt.Println(kelvin,"is K and ",celsius," is C")
    //output: 294is K and 20.85000000000023 is C
}
```

JULIA PROGRAMMING LANGUAGE.

MULTI-PARADIGM: MULTIPLE DISPATCH (PRIMARY PARADIGM), PROCEDURAL, FUNCTIONAL, META, MULTISTAGED

Julia is a high-level, high-performance, dynamic programming language. While it is a general purpose language and can be used to write any application, many of its features are well-suited for numerical analysis and computational science. Distinctive aspects of Julia's design include a type system with parametric polymorphism in a dynamic programming language; with multiple dispatch as its core programming paradigm. Julia supports concurrent, (composable) parallel and distributed computing (with or without using MPI and/or the built-in corresponding clarification needed to "OpenMP-style" threads, and direct calling of C and Fortran libraries without glue code.

DYLAN, LISP, LUA, MATHEMATICA
MATLAB, PERL, PYTHON, R, C
RUBY, SCHEME

NAME

— IN THIS LANGUAGE —

```
name1="Muhammad Allah Rakha"  
println(name1)  
#output: Muhammad Allah Rakha  
println(typeof(name1))  
#output: String
```

JULIA

— PROGRAMMING —

```
#DATA TYPE  
num1= Int64(14)  
println(num1)  
#output: 14  
num2=typeof(num1)  
println(num2)  
#output: Int64  
num3=333  
println(num3)  
#output: 333  
println(typeof(num3))  
#output: Int64  
num4=32.22  
println(num4)  
#output: 32.22  
println(typeof(num4))  
#output: Float64  
for i in 1:5  
    println(i)  
end  
#output: 1 2 3 4 5  
name1=[3,5,6,7,2,4,2,3]  
for i in name1  
    if isodd(i)  
        println("Odd \n")  
    else  
        println("Even \n")  
    end  
end  
#output: Odd Odd Even Odd  
#      Even Even Even Odd
```

AAAASSSTARBOOK OF CODING

AAAASSTAAARK

```
num1=12;
if num1>10
    println("Big")
elseif num1<13
    println("Small")
else
    println("Negative")
end
#output: Big
name1=10; name2=15;
while name1<=name2
    println(name1)
    name1+=1
end
#output: 10 11 12 13 14 15
name1=10; name2=150;
while name1<=name2
    println(name1)
    name1+=1
    if(name1>=50)
        break
    end
end
#output: 10 11 12 to 48 49
name1=10; name2=20;
while name1<=name2
    println(name1)
    name1+=1
    if(name1>=15)
        continue
    end
end
#output: 11 12 13 14
#      16 17 18 19 20
name1=[1,2,3,4,5]
println(name1)
#output: [1, 2, 3, 4, 5]
for i in name1
    println(i," ")
end
#output: 1 2 3 4 5
```

```

#COMPLEX NUMBERS
name1=1+4im # im is imagery
println(name1)
#output: 1+4im
println(real(name1))
#output: 1
println(imag(name1))
#output: 4
name2=2.3
name3=-3.4
name4=complex(name2,name3)
println(name4)
#output: 2.3 - 3.4im
name5=name4
name6=abs(name5)
println(name6)
#output: 4.104875150354758
name7=abs2(name6)
# squared of name6
println(name7)
#output: 16.849999999999994
name8=angle(name7)
println(name8)
#output: 0.0
name9=conj(name4)#2.3-3.4im
println(name9) #conjugate
#output: 2.3 + 3.4im
#REGULAR EXPRESSIONS
name2=r"( .+)@(.+)"
name3="p190006@nu.edu.pk"
name4=match(name2,name3)
println(name4)
#output: RegexMatch
#("p190006@nu.edu.pk", 1="p190006", 2="nu.edu.pk")
name5=split("Muhammad Allah Rakha",",")
println(name5)
#output: SubString{String}["Muhammad Allah Rakha"]
println(typeof(name5))
#output: Array{SubString{String},1}
a=ComplexF64[sqrt(k)+(2k)im for k in 1:5]
println(a)
#output: Complex{Float64}[1.0 + 2.0im,
# 1.4142135623730951 + 4.0im,
# 1.7320508075688772 + 6.0im,
# 2.0 + 8.0im, 2.23606797749979 + 10.0im]

```

```
name1=[2,4,55,78]
name2=copy(name1)
name3=deepcopy(name1)
println(name2,name3)
#output: [2, 4, 55, 78][2, 4, 55, 78]
name4=[sqrt(name1[i]) for i=1:length(name1)]
println("\n",name4)
#output: [1.4142135623730951, 2.0,
# 7.416198487095663, 8.831760866327848]

name6=Dict(1=>"aaaa",5=>"1234","name5"=>"M.A.R")
println(name6[1]," ",name6["name5"],name6[5])
#output: aaaa M.A.R 1234
name7=keys(name6)
name8=values(name6)
println(name7,name8)
#output: Any["name5", 5, 1]["M.A.R", "1234", "aaaa"]
```

#MATHEMATICAL FUNCTIONS

```
n1=div(10,2) #simple division
println(n1)
#output: 5
n2=fld(15,4) #floor
println(n2)
#output: 3
n3=cld(15,4) #ceil division
println(n3)
#output: 4
n4=rem(5,3) #remainder
println(n4)
#output: 2
n5=mod(5,2) #modulo
println(n5)
#output: 1
n6=mod1(6,2) #modulo with
# offset 1
println(n6)
#output: 2
```



A ~~AA~~AS T A R K

BOOK OF CODING

CPP PROGRAMMING LANGUAGE.

MULTI-PARADIGM: PROCEDURAL, FUNCTIONAL, OBJECT-ORIENTED, GENERIC

C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or "C with Classes". The language has expanded significantly over time, and modern C++ now has object-oriented, generic, and functional features in addition to facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers, including the Free Software Foundation, LLVM, Microsoft, Intel, Oracle, and IBM, so it is available on many platforms.

**ALGOL
68, C, CLU, ML, MESA, MODULA-
2, SIMULA, SMALLTALK, ADA
95, C#, C99, CHAPEL,
CLOJURE, D, JAVA, JS++,
LUA, NIM, PERL, PHP, PYTHON,
RUST, SEED7**

NAME

IN THIS LANGUAGE

```
#include <fstream>
#include <string>
#include <cstring>
#include <cmath>
#include <ctime>
#include <cstdlib>
using namespace std;
int main(int argc,char** argv){

    cout<<"Muhammad Allah Rakha"=><endl;
    //output: Muhammad Allah Rakha

    string name;
    cout<<"Enter your name : ";
    getline(cin,name);
    //enter: AAAA STARK

    cout<<"Your name is "<<name<<endl;
    //output: AAAA STARK

    string name1;
    // #include <string> is must for (getline)
    cout<<"Enter your name:";
    getline(cin,name1);
    //enter: Enter your name: Muhammad Allah Rakha

    cout<<"Name is "<<name1;
    //output: Name is Muhamma Allah Rakha

    return 0;
}
```

BOOK OF CODING
AAAAA STARK

C++

PROGRAMMING

```
#include <iostream>
#include <string>
using namespace std;
int main(int argc,char** argv){
    int num1=10;
    cout<<"num1 : "<<num1<<endl;
    //output: num1 : 10
    num1++;
    cout<<"num1++ : "<<num1<<endl;
    //output: num1++ : 11
    --num1;
    cout<<"--num1 : "<<num1<<endl;
    //output: --num1 : 10

    int n1,n2; n1=10;n2=20;
    cout<<"n1: "<<n1<<endl;
    //output: n1: 10
    cout<<"n2: "<<n2<<endl;
    //output: n2: 20
    char c1,c2,c3; c1='M';c2='A';c3='R';
    cout<<"c1: "<<c1<<" "<<"c2: "<<c2<<" ";
    cout<<"c3: "<<c3<<endl;
    //output: c1: M c2: A c3: R
    float f1,f2; f1=12.34; f2=56.78;
    cout<<"f1: "<<f1<<" "<<"f2: "<<f2<<endl;
    //output: f1: 12.34 f2: 56.78
    double d1,d2; d1=123.456789;d2=987.654321;
    cout<<"d1: "<<d1<<" "<<"d2: "<<d2<<endl;
    //output: d1: 123.456789 d2: 987.654321

    int n1,n2,n3;
    n1=10;n2=20;
    n3=n2/n1;
    cout<<"n3: "<<n3<<endl;
    //output: n3: 2
    n3=n2*n1;
    cout<<"n3: "<<n3<<endl;
    //output: n3: 200
    n3=n2+n1;
    cout<<"n3: "<<n3<<endl;
    //output: n3: 30
    n3=n2-n1;
    cout<<"n3: "<<n3<<endl;
    //output: n3: 10
    return 0;
}
```

BOOK OF CODING
AAAASSTARK



AAAASSTAAARK

BOOK OF CODING

```
#include <iostream>
#include <cstring>
#include <iomanip>
#include <string>
using namespace std;
int main(int argc, char** argv) {

    char name1[15] = "Muhammad";
    char name2[15] = "Allah";
    char name3[15] = "Rakha";
    char name4[15] = "Nacih";
    cout << name1 << " " << name2 << " ";
    cout << name3 << " " << name4 << endl;
    //output: Muhammad Allah Rakha Nacih

    char name5[100];

    strcpy(name5, name1);

    cout << name5 << " ";

    strcpy(name5, name2);

    cout << name5 << " ";

    strcpy(name5, name3);

    cout << name5 << " ";

    strcpy(name5, name4);

    cout << name5 << endl;
    //output: Muhammad Allah Rakha Naich

    string n1, n2, n3; n1 = "M"; n2 = "A"; n3 = "R";

    string n4 = n1 + n2 + n3;

    cout << "Name is " << n4 << endl;
    //output: Name is MAR
    return 0;
}
```

BOOK OF CODING

AAAAA STARK

```
#include <iostream>
using namespace std;
#define NAME1 "AAAA"
#define NAME2 "STARK"
#define NUM1 5
#define NUM2 10
int main(int argc,char** argv){
    cout<<NAME1<<" "<<NAME2;
    //output: AAAA STARK
    int number;
    number=NUM1+NUM2;
    cout<<"Number: "<<number<<endl;
    //output: Number: 15

    const string name1="Muhammad";
    const string name2="Allah";
    const string name3="Rakah";
    cout<<name1<<" "<<name2<<" "<<name3<<endl;
    //output: Muhammad Allah Rakah

    const int num1=10;
    const int num2=15;
    int sum=num1+num2;
    cout<<"num1 + num2 = "<<sum<<endl;
    //output: num1 + num2 = 25

    static int num3;
    num3=1; num3++;
    cout<<"num3: "<<num3<<endl;
    //output: 2
    num3++;
    cout<<"num3: "<<num3<<endl;
    //output: 3
    num3++;
    cout<<"num3: "<<num3<<endl;
    //output: 4

    static string name4;
    name4="M.A.R";
    cout<<"name4: "<<name4<<endl;
    //output: name4: M.A.R
    name4="AAAA STARK";
    cout<<"name4: "<<name4<<endl;
    //output: name4: AAAA STARK
    return 0;
}
```

BOOK OF CODING

AAAASSTAAARK

```
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    int num1=10;
    if(num1<10){
        cout<<"1: work";
    }
    else{
        cout<<"2: work";
    }
    //output: 2: work
    if(num1>10){
        cout<<"1: work";
    }else if(num1!=10){
        cout<<"2: work";
    }else if(num1==10){
        cout<<"3: work";
    }
    //output: 3: work

    bool b1=true;
    bool b2=false;
    if (b1 && b2){
        cout<<"true && false = true";
    }else{
        cout<<"true && false = false";
    }
    //output: true && false = false
    if (b1 || b2){
        cout<<"true || false = true";
    }else{
        cout<<"true || false =false";
    }
    //output: true || false = true
    if(b1 && (!b2)){
        cout<<"true && !false = true";
    }
    //output: true && !false = true

    return 0;
}
```

BOOK OF CODING

AAAS TARK

```
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    int num1=10;
    while(num1<=15){
        cout<<num1<<" ";
        num1+=1;
    }
    //output: 10 11 12 13 14 15
    do{
        cout<<num1<<" ";
        num1++;
    }while(num1<=20);

    for(int i=0;i<=5;i++){
        cout<<i<<" ";
    }
    //output: 16 17 18 19 20

    //output: 0 1 2 3 4 5
    for(int i=0;i<4;i++){
        for(int j=0;j<4;j++){
            cout<<i<<" "<<j<<" ";
        }
    }
    //output: 0 0 0 1 0 2 0 3 1
    //          0 1 1 1 2 1 3 2 0
    //          2 1 2 2 2 3 3 0 3
    //          1 3 2 3 3

    int num1=10;
    while(num1<=15){
        cout<<num1<<" ";
        if(num1>=13){
            break;
        }
        num1+=1;
    }
    //output: 10 11 12 13

    int num1=10;
    while(num1<=15){
        if(num1==13){
            num1+=1;
            continue;
        }
        cout<<num1<<" ";
        num1+=1;
    }
    //output: 10 11 12 14 15
    return 0;
}
```



BOOK OF CODING

AIAA STARK

```
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    char name;
    cout<<"Enter a Oner Caharacter.";
    //enter: A

    cin>>name;
    switch (name){
        case 'A':
            cout<<"A:";
            break;
        case 'B':
            cout<<"B:";
            break;
        default:
            cout<<"Invalid Caharacter.";
    }
    //output: :A:

    char name1;
    cout<<"Enter a Oner Caharacter.";
    //enter: a
    cin>>name1;

    switch (name1){
        case 'A':
            cout<<"A:";
            break;
        case 'B':
            cout<<"B:";
            break;
        default:
            cout<<"Invalid Caharacter.";
    }
    //output: Invalid Caharacter.

    return 0;
}
```

```
#include <iostream>
#include <ctime>
#include <cstdlib>
using namespace std;
int main(int argc, char** argv) {
    srand(time(NULL));
    for(int i=0;i<=5;i++){
        cout<<"Ranandom Number ["<<i<<"] ="<<rand()<<endl;
    }
    //output: Ranandom Number [0] =5765
    //          Ranandom Number [1] =20153
    //          Ranandom Number [2] =29153
    //          Ranandom Number [3] =14284
    //          Ranandom Number [4] =24536
    //          Ranandom Number [5] =8829
    return 0;
}
```

```
#include <iostream>
#include <cstring>
#include <iomanip>
using namespace std;
int main(int argc, char** argv) {
    int num=10;
    cout<<"Number_simple : "<<num<<endl;
    //output: Number_simple : 10

    int *num1;
    num1=&num;
    cout<<"Number_pointer : "<<*num1<<endl;
    //output: Number_pointer : 10

    int **num2;
    num2=&num1; //here address from previous pointer
    cout<<"Number_double_pointer: "<<**num2<<endl;
    //output: Number_double_pointer: 10

    return 0;
}
```

```
#include <iostream>
#include <string>
using namespace std;
// upper function create now
void Name1(){
// here function is declear
    cout<<"Enter your name! ";
//enter: Enter your name! AAAA STARK
    string name;
    getline(cin,name);
    cout<<"Name is "<<name<<endl;
//output: Name is AAAA STARK
}
int main(int argc, char** argv) {
// here function is call
    Name1();
    return 0;
}
```

```
#include <iostream>
#include <string>
using namespace std;
void Name1();
// below function is create here is
// defined now with same function
// handing and ;
int main(int argc, char** argv) {
// here function is call
    Name1();
    return 0;
}
void Name1(){
// here function is declear
    cout<<"Enter your name! ";
//enter: Enter your name! AAAA STARK
    string name;
    getline(cin,name);
    cout<<"Name is "<<name<<endl;
//output: Name is AAAA STARK
}
```



AAAAA STARK

BOOK OF CODING

```
#include <iostream>
#include <string>
using namespace std;
string Name1();
int Name2();
int main(int argc, char** argv) {
    string names=Name1();
    cout<<"Your name is "<<names<<endl;
    //output: Your name is AAAA STARK
    int numbers=Name2();
    cout<<"Your age is "<<numbers<<endl;
    //output: Your age is 20
    return 0;
}
string Name1(){
// when you return anyone then
// return_type is not (void).
// here I am return_string now
// return_type is (string).
    string name;
    cout<<"Enter your name!";
    //enter: Enter your name!AAAA STARK
    getline(cin,name);
    return name;
}
int Name2(){
// when you return anyone then
// return_type is not (void).
// here I am return_string now
// return_type is (string).
    int num;
    cout<<"Enter your age!";
    cin>>num;
    //enter: Enter your age!20
    return num;
}
```

```
#include <iostream>
#include <string>
using namespace std;
void Name3(int n,string s);
int Name4(int n1,int n2);
void Name5(int *n,string *s);
int main(int argc, char** argv) {
    int a=20;
    string n="M.A.R";
    Name3(a,n);

    int numbers=Name4(5,10);
    cout<<"n1 + n2 ="<<numbers;
    //output: n1 + n2 =15

    int n1=20; string s1="M.A.R";
    Name5(&n1,&s1);
    // here pass (address & )
    return 0;
}
void Name3(int n,string s){
    cout<<"Name : "<<s<<endl;
    //output: Name : M.A.R
    cout<<"Age : "<<n<<endl;
    //output: Age : 20
}
int Name4(int n1,int n2){
    cout<<"Two number is add"<<endl;
    //output: Two number is add
    return (n1+n2);
}
void Name5(int *n,string *s){
// here point of address by using
// (pointer *).
    cout<<"Name is "<<*s<<endl;
    //output: Name is M.A.R
    cout<<"Age is "<<*n<<endl;
    //output: Age is 20
}
```

BOOK OF CODING

ARRAYS

```
//Arrays

#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    int name1[]={1,2,3,4,5,6,7,8,9,10};
    for(int i=0;i<10;i++){
        cout<<name1[i]<<" ";
    }
    //output: 1 2 3 4 5 6 7 8 9 10
    cout<<"name1[0] ="<<name1[0]<<endl;
    //output: 1

    int name2[10]={1,2,3,4,5,6,7,8,9,10};
    for(int i=0;i<10;i++){
        cout<<name2[i]<<" ";
    }
    //output: 1 2 3 4 5 6 7 8 9 10
    cout<<"name2[9] ="<<name2[9]<<endl;
    //output: 10

    string names[4]={};
    string name;
    for(int i=0;i<4;i++){
        cout<<"Enter a Name: ["<<i<<"] =";
        cin>>name;
        names[i]=name;
    }
    //enter: Enter a Name: [0] =Muhammad
    //        Enter a Name: [1] =Allah
    //        Enter a Name: [2] =Rakha
    //        Enter a Name: [3] =Naich
    for(int i=0;i<4;i++){
        cout<<"Your Name : ["<<i<<"] =";
        cout<<names[i]<<endl;;
    }
    //output: Your Name : [0] =Muhammad
    //        Your Name : [1] =Allah
    //        Your Name : [2] =Rakha
    //        Your Name : [3] =Naich
    return 0;
}
```

BOOK OF CODING

BOOK OF STARK

```
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    int name[4][4]={
        {1,2,3,4},
        {5,6,7,8},
        {9,10,11,12},
        {13,14,15,16}
    };
    for(int i=0;i<4;i++){
        for(int j=0;j<4;j++){
            cout<<"name["<<i<<"]["<<j<<"] =";
            cout<<name[i][j]<<endl;
        }
    }
    //output: name[0][0] =1 name[0][1] =2
    //         name[0][2] =3 name[0][3] =4
    //         name[1][0] =5 name[1][1] =6
    //         name[1][2] =7 name[1][3] =8
    //         name[2][0] =9 name[2][1] =10
    //         name[2][2] =11 name[2][3] =12
    //         name[3][0] =13 name[3][1] =14
    //         name[3][2] =15 name[3][3] =16
    string names[4]={"Muhammad", "Allah", "Rakha", "Naich"};
    string *name_pointer;
    name_pointer=names;
    for(int i=0;i<4;i++){
        cout<<"*(name_pointer +<<i<<") =";
        cout<<*(name_pointer+i)<<endl;
    }
    //output: *(name_pointer +0) =Muhammad
    //output: *(name_pointer +1) =Allah
    //output: *(name_pointer +2) =Rakha
    //output: *(name_pointer +3) =Naich
    for(int i=0;i<4;i++){
        cout<<"*(names +<<i<<") =";
        cout<<*(names + i)<<endl;
    }
    //output:      *(names +0) =Muhammad
    //output:      *(names +1) =Allah
    //output:      *(names +2) =Rakha
    //output:      *(names +3) =Naich
    return 0;
}
```

```
#include <iostream>
using namespace std;
void Name_Array1(int anyone[],int limit);
int Name_Array1_sum(int anyone[],int limit);
int main(int argc, char** argv) {
    int array1[5]={1,2,3,4,5};
    Name_Array1(array1,5);
    //output: anyone [0]=1
    //          anyone [1]=2
    //          anyone [2]=3
    //          anyone [3]=4
    //          anyone [4]=5
    int sum_return=Name_Array1_sum(array1,5);
    cout<<"Sum of Array = "<<sum_return<<endl;
    //output: Sum of Array = 15
    return 0;
}
void Name_Array1(int anyone[],int limit){
    for(int i=0;i<limit;i++){
        cout<<"anyone ["<<i<<"]=";
        cout<<anyone[i]<<endl;
    }
}
int Name_Array1_sum(int anyone[],int limit){
    int sum=0;
    for(int i=0;i<limit;i++){
        sum+=anyone[i];
    }
    return sum;
}
```

AAAAS STARK

```
//File I/O  
#include <iostream>  
#include <fstream>  
using namespace std;  
int main(){  
    // Muhammad ALLah Rakha  
    // Naich and AAAA STARK  
    // My (stark.txt) file data.  
    fstream file1; char name2[100];  
    file1.open("stark.txt",ios::in);  
    file1.getline(name2,100);  
    cout<<"1: "<<name2<<endl;  
    //output: 1: Muhammad ALLah Rakha  
    file1.getline(name2,100);  
    cout<<"2: "<<name2<<endl;  
    //output: 2: Naich and AAAA STARK  
    file1.close();  
    //here not to be access of File_Data  
    return 0; }  
  
#include <iostream>  
#include <fstream>  
using namespace std;  
int main(){  
    // Write data file.  
    // My (stark.txt) file data.  
    fstream file2; string name1;  
    file2.open("stark.txt",ios::out);  
    cout<<"Enter a Name :1: ";  
    getline(cin,name1);  
    //enter: Muhammad ALLah Rakha  
    file2<<name1; file2<<"\n";  
    cout<<"Enter a Name :2: ";  
    getline(cin,name1);  
    //enter: Naich AAAA STARK  
    file2<<name1;  
    file2.close();  
  
    // Read data file.  
    // My (stark.txt) file data.  
    fstream file1; char name2[100];  
    file1.open("stark.txt",ios::in);  
    file1.getline(name2,100);  
    cout<<"1: "<<name2<<endl;  
    //output: 1: Muhammad ALLah Rakha  
    file1.getline(name2,100);  
    cout<<"2: "<<name2<<endl;  
    //output: 2: Naich AAAA STARK  
    file1.close();  
    //here not to be access of File_Data  
    return 0; }
```

AAAASSTARK

```
//Exception Handling  
#include <iostream>  
#include <fstream>  
using namespace std;  
int main(){  
    string name1;  
    string name2;  
    string name3;  
    name1="AAAA STARK";  
    name2="aaaa stark";  
    try{  
        name3="You know both are not equal.";  
        if(name1==name2){  
            cout<<"1: is work.";  
        }else{  
            throw name3;  
        }  
        // three (...) means. Here every data type is throw  
        // e.g. (string,int,double,float,etc.)  
    }catch(...){  
        cout<<name3<<endl;  
        //output: You know both are not equal.  
    }  
  
    int num1=10;  
    int num2=20;  
    int sum=0;  
    try{  
        if(num1!=num2){  
            sum=num1+num2;  
            throw sum;  
        }  
    }catch(...){  
        cout<<"Sum of two number : "<<sum<<endl;  
        //output: Sum of two number : 30  
    }  
    return 0;  
}
```

//Data and Time

```
#include <iostream>
#include <ctime>
using namespace std;
int main(int argc, char** argv) {
    cout<<"Time and data based on current system.\n";
    //output: Time and data based on current system.
    time_t now = time(0);
    char* name1=ctime(&now);
    cout<<"The Local Data and Time is "<<name1<<endl;
    //output: The Local Data and Time is Sun Sep 20 23:36:52 2020
    cout<<"The convert now to TM struct for UTC.\n";
    //output: The convert now to TM struct for UTC.
    tm *gmtm=gmtime(&now);
    name1=asctime(gmtm);
    cout<<"The UTC data and Time is "<<name1<<endl;
    //output: The UTC data and Time is Sun Sep 20 18:39:56 2020
    return 0;
}
```

```
#include <iostream>
#include <string>
#include <cmath>
using namespace std;
int main(int argc, char** argv) {
    int n1,n2;
    n1=10;n2=-20;
    double d1=123.4566;
    cout<<"abs(n2) ="<<abs(n2)<<endl;
    //output: abs(n2) =20
    cout<<"floor(d1) ="<<floor(d1)<<endl;
    //output: floor(d1) =123
    cout<<"sqrt(n1) ="<<sqrt(n1)<<endl;
    //output: sqrt(n1) =3.16228
    cout<<"pow(n1,2)  ="<<pow(n1,2)<<endl;
    //output: sqrt(n1) =3.16228
    return 0;
}
```

BOOK OF CODING

AAAASSTARTRK

```
//Structure
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
struct Name1{
    string name;
    int age;
    char name1[15];
};
struct Name2{
    string name;
    int age;
    void set_values(int n,string s){
        name=s; age=n;
    }
    int age_get(){
        return age;
    }
    string name_get(){
        return name;
    }
};
int main(int argc, char** argv) {
    Name1 name1;
    name1.age=20;
    strcpy(name1.name1,"AAAA STARK");
    name1.name="Muhammad Allah Rakha";
    cout<<"Name: "<<name1.name<<endl;
    //output: Name: Muhammad Allah Rakha
    cout<<"Age : "<<name1.age<<endl;
    //output: Age : 20
    cout<<"Industry: "<<name1.name1<<endl;
    //output: Industry: AAAA STARK

    Name2 name2;
    name2.set_values(20,"M.A.R");
    cout<<"Name: "<<name2.name_get()<<endl;
    //output: Name: M.A.R
    cout<<"Age: "<<name2.age_get()<<endl;
    //output: Age: 20
    return 0;
}
```

```
//NameSpaces
#include <iostream>
using namespace std;
namespace name1{
    void name(){
        string s;
        s="Muhammad Allah Rakha";
        cout<<"Name : "<<s;
//output: Name : Muhammad ALLah Rakha
    }
}
namespace name2{
    void age(int n){
        cout<<"Age : "<<n;
        //output: Age : 20
    }
}
int main(){
    name1::name();
    cout<<endl;
    name2::age(20);
    return 0;
}
```

```
#include <iostream>
using namespace std;
namespace name1{
    void name(){
        string s;
        s="Muhammad Allah Rakha";
        cout<<"Name : "<<s;
//output: Name : Muhammad ALLah Rakha
    }
}
namespace name2{
    void age(int n){
        cout<<"Age : "<<n;
        //output: Age : 20
    }
}
using namespace name1;
using namespace name1::name2;
int main(){
    name1::name();
    cout<<endl;
    age(20);
    return 0;
}
```

AAAASSTARK

BOOK OF CODING

```
//Class and Objects

#include <iostream>
using namespace std;
class name{ //here object.
// e.g (name,Name,nameOf,etc)
public:
    string name;
    int age;
    void set_age(int a){
        age=a;
    }
    int get_age(){
        return age;
    }
};

int main(int argc, char** argv) {
    name n1;
    n1.name="AAAA STARK";
    cout<<"Name is : "<<n1.name<<endl;
    //output: Name is : AAAA STARK
    n1.set_age(20);
    cout<<"Age is : "<<n1.get_age()<<endl;
    //output: Age is : 20

    name n2;
    n2.name="Muhammad Allah Rakha";
    cout<<"Name is :"<<n2.name<<endl;
    //output: Name is :Muhammad Allah Rakha
    n2.set_age(20);
    cout<<"Age is : "<<n2.get_age()<<endl;
    //output: Age is : 20
    return 0;
}
```

BOOK OF CODING

AAAAS TARK

```
#include <iostream>
using namespace std;
class name{ //here object.
// e.g (name,Name,nameOf,etc)
private:
    int age;
    string name1;
public:
    string name;
    void set_age(int a){
        age=a;
    }
    int get_age(){
        return age;
    }
    void set_name1(string n){
        name1=n;
    }
    string get_name1(){
        return name1;
    }
};
int main(int argc, char** argv) {
    name n1;
    n1.name="AAAA STARK";
    cout<<"Name is : "<<n1.name<<endl;
    //output: Name is : AAAA STARK
    n1.set_age(20);
    cout<<"Age is : "<<n1.get_age()<<endl;
    //output: Age is : 20

    name n2;
    //here (n2) is use just call private.
    n2.set_name1("M.A.R");
    cout<<"Name is now: "<<n2.get_name1()<<endl;
    //output: Name is now: M.A.R
    n2.set_age(20);
    cout<<"Age is now:"<<n2.get_age()<<endl;
    //output: Age is now:20
    return 0;
}
```

```
class Me{ //here object.  
// e.g (me,Me,meOf,etc)  
private:  
    int age;  
    string name;  
public:  
    void set_name_age(string s,int n);  
    string get_name();  
    int get_age();  
};  
void Me::set_name_age(string s,int n){  
    age=n; name=s;  
}  
  
string Me::get_name(){  
    return name;  
}  
  
int Me::get_age(){  
    return age;  
}  
int main(int argc, char** argv) {  
    Me me1;  
    me1.set_name_age("M.A.R",20);  
    cout<<"Name : "<<me1.get_name()<<endl;  
    //output: Name : M.A.R  
    cout<<"Age : "<<me1.get_age()<<endl;  
    //output: Age : 20  
    return 0;  
}
```

AAAAA STARK

```
#include <iostream>
using namespace std;
class My{//here object.
// e.g (my,My,myOf,etc)
protected:
    string anyname;
    double anynum;
};
class Me:My{ //here object.
// e.g (me,Me,meOf,etc)
public:
    void set_name_age(string s,int n);
    string get_name();
    int get_age();
};

void Me::set_name_age(string s,int n){
    anynum=n; anyname=s;
}
string Me::get_name(){
    return anyname;
}
int Me::get_age(){
    return anynum;
}
int main(int argc, char** argv) {
    Me me1;
    me1.set_name_age("",20);
    cout<<"Name : "<<me1.get_name()<<endl;
    //output: Name : AAAA STARK
    cout<<"Age : "<<me1.get_age()<<endl;
    //output: Age : 20
    return 0;
}
```

```
#include <iostream>
using namespace std;
class Me{ //here object.
// e.g (me,Me,meOf,etc)
private:
    int age;
    string name;
public:
// here call a constructor
// You can use two time constructor
// or many time.
    Me(int n,string s){
        age=n; name=s;
    }
    void set_name_age(string s,int n);
    string get_name();
    int get_age();
// here call a distructer
// when constructor call two time then
// distructer automatically call two time.
    ~Me(){
        cout<<"Also Address is clear.";
    }
};

void Me::set_name_age(string s,int n){
    age=n; name=s;
}

string Me::get_name(){
    return name;
}

int Me::get_age(){
    return age;
}

int main(int argc, char** argv) {
    Me me1(20,"M.A.R");
    cout<<"My name: "<<me1.get_name()<<endl;
    //output: My name: M.A.R
    cout<<"My age: "<<me1.get_age()<<endl;
    //output: My age: 20

    me1.set_name_age("AAAA STARK",20);
    cout<<"Name : "<<me1.get_name()<<endl;
    //output: Name : AAAA STARK
    cout<<"Age : "<<me1.get_age()<<endl;
    //output: Age : 20

    //output: Also Address is clear.
    return 0;
}
```

BOOK OF CODING

AAAAA STARK

```
#include <iostream>
using namespace std;
class Me{ //here object.
// e.g (me,Me,meOf,etc)
private:
    string name;
    int age;
public:
    Me(int n,string s){
        name=s; age=n;
    }
    string get_name(){
        return name;
    }
    int get_age(){
        return age;
    }
    ~Me(){
        cout<<"Also Address are clear.";
    }
}
int main(int argc, char** argv) {
    Me me1(20,"AAAA STARK"); // distructer is auto call.
    cout<<"Name : "<<me1.get_name()<<endl;
    //output: Name : AAAA STARK
    cout<<"Age : "<<me1.get_age()<<endl;
    //output: Age : 20

    //Also Address are clear.

    return 0;
}
```

```
#include <iostream>
using namespace std;
class Me{ //here object.
// e.g (me,Me,meOf,etc)
private:
    int *pointer;
public:
    int get_age();
    Me(int age);
    Me(const Me &anynum);
    ~Me();
};

Me::Me(int age){
    cout<<"Normal constructer";
    cout<<" allocate memory for Pointer" << endl;
//output: Normal constructer allocate memory for Pointer
    pointer = new int;
    *pointer=age;
}

Me::Me(const Me &anynum){
// Copy constructer allocate Pointer.
//here is not allow (cout)
    pointer =new int;
    *pointer = *anynum.pointer;
}

Me::~Me(){
    cout<<"Auto Call and Freeing Memory.\n";
//output: Auto Call and Freeing Memory.
//      Auto Call and Freeing Memory.
    delete pointer;
}

Me::get_age(){
    return *pointer;
}

int main(int argc, char** argv) {
    Me me1(20);
    cout<<"My Age : (constructer) ";
    cout<<me1.get_age()<< endl;
//output: My Age : (constructer) 20
    Me me2=me1; // this is COPY CONSTRUCTER
    cout<<"My Age : (copy constructer) ";
    cout<<me2.get_age()<< endl;
//output: My Age : (copy constructer) 20
    return 0;
}
```

```
#include <iostream>
using namespace std;
class Me{ //here object.
// e.g (me,Me,meOf,etc)
    private:
        string *pointer;
    public:
        string get_name(){
            return *pointer;
        };
        Me(string name);
        Me(const Me &anyname);
        ~Me();
};

Me::Me(string name){
    cout<<"Normal constructer";
    cout<<" allocate memory for Pointer"<<endl;
//output: Normal constructer allocate memory for Pointer
    pointer = new string;
    *pointer=name;
}

Me::Me(const Me &anyname){
// Copy constructer allocate Pointer.
//here is not allow (cout)
    pointer =new string;
    *pointer = *anyname.pointer;
}

Me::~Me(){
    cout<<"Auto Call and Freeing Memory.\n";
//output: Auto Call and Freeing Memory.
//      Auto Call and Freeing Memory.
    delete pointer;
}

//Me::get_name(){
//    return *pointer;
//}

int main(int argc, char** argv) {
    Me me1("Muhamma Allah Rakha");
    cout<<"My Name : (constructer) ";
    cout<<me1.get_name()<<endl;
//output: My Name : (constructer) Muhamma Allah Rakha
    Me me2=me1; // this is COPY CONSTRUCTER
    cout<<"My Name : (copy constructer) ";
    cout<<me2.get_name()<<endl;
//output: My Name : (copy constructer) Muhamma Allah Rakha

    return 0;
}
```

```
#include <iostream>
using namespace std;
class Me{ //here object.
// e.g (me,Me,meOf,etc)
private:
    string name;
    int age;
public:
    void set_name(string s);
    void set_age(int n);
    friend void print_name(Me me1);
    friend void print_age(Me me3);
};

void Me::set_name(string s){
    name=s;
}

void Me::set_age(int n){
    age=n;
}

void print_name(Me me1){
    cout<<"Name : "<<me1.name<<endl;
    //output: Name : M.A.R
}

void print_age(Me me3){
    cout<<"Age : "<<me3.age<<endl;
    //output: Age : 20
}

int main(int argc, char** argv) {
    Me me2;
    me2.set_name("M.A.R");
    print_name(me2);
    me2.set_age(20);
    print_age(me2);
    return 0;
}
```

BOOK OF CODING

```
#include <iostream>
using namespace std;
class Me{ //here object.
// e.g (me,Me,meOf,etc)
private:
    string name;
    int age;
public:
    Me(string s,int n){
        name=s; age=n;
    }
    string get_name(){
        return name;
    }
    int get_age(){
        return age;
    }
};
int main(int argc, char** argv) {
    Me me1("M.A.R",20);
    Me *me;
    me=&me1;
    cout<<"My Name is "<<me->get_name()<<endl;
    //output: My Name is M.A.R
    cout<<"My Age is "<<me->get_age()<<endl;
    //output: My Age is 20
    return 0;
}

#include <iostream>
using namespace std;
class Me{ //here object.
// e.g (me,Me,meOf,etc)
private:
    string name;
public:
    static int count;
    Me(string s){
        name=s;
        count++;
    }
    string get_name(){
        return name;
    }
};
int Me::count=0;
int main(int argc, char** argv) {
    Me me1("M.A.R");
    cout<<"My Name is "<<me1.get_name()<<endl;
    //output: My Name is M.A.R
    cout<<"Total Objects counting : "<<Me::count<<endl;
    //output: Total Objects counting : 1
    return 0;
}
```

```
//Inheritance

#include <iostream>
using namespace std;
class Me{ //here object.
// e.g (me,Me,meOf,etc)
protected:
    string name;
public:
    void set_name(string s){
        name=s;
    }
};

class Me2: public Me{
// here considered a public.
public:
    string get_name(){
        return name;
    }
};

int main(int argc, char** argv) {
    Me2 me2;
    me2.set_name("Muhammah Allah Rakha");
    string anyname=me2.get_name();
    cout<<"My name is "<<anyname<<endl;
    //output: My name is Muhammah Allah Rakha
    return 0;
}
```



AAAASSТАРК

BOOK OF CODING

```
#include <iostream>
using namespace std;
class Me{ //here object.
// e.g (me,Me,meOf,etc)
protected:
    string name;
public:
    void set_name(string s){
        name=s;
    }
};
class Me2{
public:
    int get_age(int a){
        return a;
    }
};
class Me3: public Me, public Me2{
// here considered a public.
public:
    string get_name(){
        return name;
    }
};
int main(int argc, char** argv) {
    Me3 me3;
    me3.set_name("Muhammad Allah Rakha");
    string anyname=me3.get_name();
    int anynum=me3.get_age(20);
    cout<<"My name is "<<anyname<<endl;
    //output: My name is Muhammad Allah Rakha
    cout<<"My age is "<<anynum<<endl;
    //output: My age is 20
    return 0;
}
```

```
//Polymorphism
#include <iostream>
using namespace std;
class Me1{
protected:
    string name1;
    int num1;
public:
    void sets0(string s,int n){
        name1=s; num1=n;
    }
    void prints0(){
        cout<<"Name (protected in Me1) : ";
        cout<<name1<<endl;
        //output: Name (protected in Me1) : M.A.R
    }
    void prints1(){
        cout<<"Name (protected in Me2) : ";
        cout<<name1<<endl;
        //output: Name (protected in Me2) : AAAA STARK
    }
    void prints0(){
        cout<<"Number (protected in Me1) : ";
        cout<<num1<<endl;
        //output: Number (protected in Me1) : 20
    }
    void prints1(){
        cout<<"Number (protected in Me2) : ";
        cout<<num1<<endl;
        //output: Number (protected in Me2) : 12345
    }
};
class Me2: public Me1{
public:
    void sets1(string s,int n){
        name1=s; num1=n;
    }
    void prints1(){
        cout<<"Name (protected in Me2) : ";
        cout<<name1<<endl;
        //output: Name (protected in Me2) : AAAA STARK
    }
    void prints0(){
        cout<<"Number (protected in Me1) : ";
        cout<<num1<<endl;
        //output: Number (protected in Me1) : 6789
    }
};
int main(){
// here Class Me1 in which not access Me2 of functions.
// this functions are Me1(Class)
    Me1 me1;
    me1.sets0("M.A.R",20);
    me1.prints0();

// here Class Me2 in which access Me1 of functions.
// this functions are Me2(Class)
    Me2 me2;
    me2.sets1("AAAA STARK",12345);
    me2.prints1();
// this functions are Me1(Class)
    me2.sets0("Muhammad Allah Rakha",6789);
    me2.prints0();
    return 0; }
```

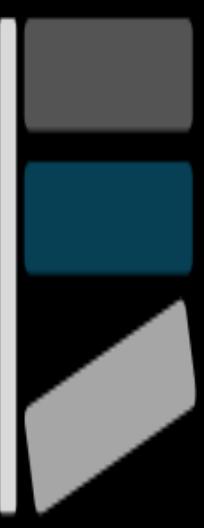
```
#include <iostream>
using namespace std;
class Me1{
protected:
    string name1;
    int num1;
public:
    void sets(string s,int n){
        name1=s; num1=n;
    }
    void prints(){
        cout<<"Name (protected in Me1) : ";
        cout<<name1<<endl;
        //output: Name (protected in Me1) : M.A.R
        cout<<"Number (protected in Me1) : ";
        cout<<num1<<endl;
        //output: Number (protected in Me1) : 20
    }
};
class Me2{
private:
    string name2;
public:
    void sets1(string s){
        name2=s;
    }
    void prints1(){
        cout<<"Name (private in Me2) : ";
        cout<<name2<<endl;
        //output: Name (private in Me2) : AAAA STARK
    }
};
class Me3: public Me2,public Me1{
public:
    void anyname(){
        cout<<"Wellcome in Polymorphism.\n";
        //output: Wellcome in Polymorphism.
    }
};
int main(){
// here Class Me1 in which not access Me2 of functions.
// this functions are Me1(Class)
    Me1 me1; me1.sets("M.A.R",20); me1.prints();
// here Class Me2 in which not access Me1 of functions.
// this functions are Me2(Class)
    Me2 me2; me2.sets1("AAAA STARK"); me2.prints1();
// here Class Me3 in which both access Me1 and Me2 of functions.
// this functions are Me3(Class)
    Me3 me3; me3.anyname();
    return 0;
}
```

AAAASSTARK



//Data Abstraction and Encapsulation

```
#include <iostream>
using namespace std;
class Me{
private:
    int num;
public:
    Me(int n1=0){
        num=n1;
    }
    void addnum(int n2){
        num+=n2;
    }
    int totalnum(){
        return num;
    }
    ~Me(){
        cout<<"Clear Memory Address.\n";
        //output: Clear Memory Address.
    }
};
int main(){
    Me me;
    me.addnum(10);
    me.addnum(20);
    me.addnum(30);
    me.addnum(40);
    cout<<"The total number : ";
    cout<<me.totalnum()<<endl;
    //output: The total number : 100
    return 0;
}
```



BOOK OF CODING

A A A A S T A R K

```
//Interfaces
#include <iostream>
using namespace std;
class Me{
protected:
    int num;
public:
    virtual int totalnum()=0;
    Me(int n1=0){
        num=n1;
    }
    void addnum(int n2){
        num+=n2;
    }
    ~Me(){
        cout<<"Clear Memory Address.\n";
        //output: Clear Memory Address.
    }
};
class Me1: public Me{
public:
    int totalnum(){
        return num;
    }
};
int main(){
    Me1 me1;
    me1.addnum(10);
    me1.addnum(20);
    me1.addnum(30);
    me1.addnum(40);

    cout<<"The total number : ";
    cout<<me1.totalnum()<<endl;
    //output: The total number : 100
    return 0;
}
```

C PROGRAMMING LANGUAGE.

**IMPERATIVE (PROCEDURAL),
STRUCTURED**

C is a general-purpose, procedural computer programming language supporting structured programming, lexical variable scope, and recursion, with a static type system. By design, C provides constructs that map efficiently to typical machine instructions. It has found lasting use in applications previously coded in assembly language. Such applications include operating systems and various application software for computer architectures that range from supercomputers to PLCs and embedded systems

NUMEROUS: AMPL, AWK, CSH, C++ , C--, C#, OBJECTIVE-C, D, GO, JAVA, JAVASCRIPT, JULIA, LIMBO, LPC, PERL, PHP, PIKE, PROCESSING, PYTHON, RUST, SEED7, VALA, VERILOG (HDL), NIM, ZIGB (BCPL, CPL), ALGOL 68, ASSEMBLY, PL/I, FORTRAN

NAME

IN THIS LANGUAGE

```
#include <stdio.h> // Basic Concept of C_Language
#include <float.h> // C Language and Cpp Language are same.
int main(){
    printf("Muhammad Allah Rakha\n");
    //output: Muhammad ALLah Rakha

    int num1;
    printf("Enter a Number in int : ");
    scanf("%d",&num1);
    //enter: Enter a Number in int : 12345
    printf("Number : %d",num1);
    //output: Number : 12345

    float num2;
    printf("Enter a Number in float : ");
    scanf("%f",&num2);
    //enter: Enter a Number in points in float : 12.3
    printf("Number in points : %f",num2);
    //output: Number in points : 12.300000

    double num3;
    printf("Enter a Number in double : ");
    scanf("%lf",&num3);
    //enter: Enter a Number in double : 123.4556
    printf("Number in double : %lf",num3);
    //output: Number in double : 123.455600

    char name1;
    printf("Enter a Character in char : ");
    scanf("%c",&name1);
    //enter: Enter a Character in char : A
    printf("Character in char : %c",name1);
    //output: Character in char : A

    char name[15];
    printf("Enter a String_Name in char : ");
    scanf("%s",&name);
    //enter: Enter a String_Name in char : AAAASTARK
    printf("String_Name in char : %s",name);
    //output: String_Name in char : AAAASTARK
    return 0;
}
```



PROGRAMMING

AAAAS T A R K
BOOK OF CODING

JAVA PROGRAMMING LANGUAGE.

MULTI-PARADIGM: GENERIC, OBJECT-ORIENTED (CLASS-BASED), IMPERATIVE, REFLECTIVE

Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture.

The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them.

As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers

ADA2005, BEANSHELL, C#, CHAPEL, CLOJURE, ECMASCIPT, FANTOM, GAMBAS, GROOVY, HACK, HAXE, J#, KOTLIN, PHP, PYTHON, SCALA, SEED7, VALA

NAME

— IN THIS LANGUAGE

BOOK OF CODING
AAAAA STARK

```
import java.util.*;  
public class JavaBook1{ // here original file name  
    public static void main (String []args){  
        System.out.print("Muhammad Allah Rakha");  
        //output: Muhammad Allah Rakha
```

```
String name="M.A.R";  
System.out.print("Name is "+name);  
//output: Name is M.A.R
```

```
Scanner user_input=new Scanner(System.in);  
System.out.print("Enter your name: ");  
String name1=user_input.nextLine();  
//enter: AAAA STARK
```

```
System.out.print("Your name is "+name1);  
//output: Your name is AAAA STARK
```

```
System.out.println("Name : \"M.A.R\" ");  
//output: Name : "M.A.R"
```

```
String name1=String.format("Muhammad"+  
    "Allah"+  
    "Rakha");  
System.out.print(name1);  
//output: MuhammadAllahRakha
```

}

}

JAVA

PROGRAMMING

```
// Varable type  
import java.util.*;  
public class JavaBook{  
    Run | Debug  
    public static void main (String []args){  
        Scanner user_input=new Scanner(System.in);  
        System.out.print("Enter a number int :");  
        Integer num1=user_input.nextInt();  
        //output: Enter a number int :12345  
        System.out.print("Number in Interger : "+num1);  
        //output: Number in Interger : 12345  
  
        System.out.print("Enter a number float :");  
        Float num2=user_input.nextFloat();  
        //output: Enter a number float :12.3456  
        System.out.print("Number in Float : "+num2);  
        //output: Number in Float : 12.3456  
  
        System.out.print("Enter a number double :");  
        Double num3=user_input.nextDouble();  
        //output: Enter a number double :1234.56789  
        System.out.print("Number in Double : "+num3);  
        //output: Number in Double : 1234.56789  
  
        System.out.print("Enter a number boolean :");  
        Boolean num4=user_input.hasNext();  
        //output: Enter a number boolean :true  
        System.out.print("Number in Boolean : "+num4);  
        //output: Number in Boolean :true  
  
        System.out.print("Enter a char : ");  
        String num5=user_input.next();  
        //enter: STARK  
        Character num6=num5.charAt(0);  
        System.out.print("Number in Caharacter : "+num6);  
        //output: Number in Caharacter : S  
    }  
}
```

BOOK OF CODING
JAVA STAR

//Data Types

```
import java.util.*;
public class JavaBook{
    Run | Debug
    public static void main(String []args){
        int a=10; int b=20;
        // Your choice to use (+,-,^,%,/ )
        // or (==,!=,>, <, >=, <=)
        // or ( &&, ||, ! )
        // or ( %=, &=, ^=, |=, *=, -=, += )
        // or Variable x= (expression) ? if true : if false
        System.out.println("a+b = "+(a+b));
        //output: a+b =30
        System.out.println("a<b "+(a<b));
        //output: a<b true
        System.out.println("!(a<b) =" +( !(a<b)));
        //output: !(a<b) =false
        System.out.println("a+=b =" +(a+=b));
        //output: a+=b =30
        System.out.println("a==b ? true : false =");
        System.out.println(a==b ? true : false);
        //output: a==b ? true : false = false
    }
}
```

BOOK OF CODING

```
import java.util.*;
public class JavaBook{
    Run | Debug
    public static void main(String []args){
        int a=20; int b=25;
        while(a<=b){
            System.out.print("value a : "+a);
            a++;
            System.out.print("\n");
        }
        //output: value a : 20
        //          value a : 21
        //          value a : 22
        //          value a : 23
        //          value a : 24
        //          value a : 25
        for(int i=0;i<=5;i++){
            System.out.println("i : "+i);
        }
        //output: i : 0
        //          i : 1
        //          i : 2
        //          i : 3
        //          i : 4
        //          i : 5
        int c=5;
        do{
            System.out.print("value a : "+c);
            c++;
            System.out.print("\n");
        }while(c<=10);
        //output: value c : 5
        //          value c : 6
        //          value c : 7
        //          value c : 8
        //          value c : 9
        //          value c : 10
    }
}
```

BOOK OF CODING

AJAVA STARTER

```
import java.util.*;
public class JavaBook{
    Run | Debug
    public static void main(String []args){
        int [] num1={1,2,3,4,5};
        for(int anyname:num1){
            if(anyname==3){
                break;
            }
            System.out.print("num1 : "+anyname+"\n");
            //output: num1 : 1
            //          num1 : 2
        }
        for(int anyname:num1){
            if(anyname==3){
                continue;
            }
            System.out.print("num1 : "+anyname+"\n");
            //output: num1 : 1
            //          num1 : 2
            //          num1 : 4
            //          num1 : 5
        }
    }
}

import java.util.*;
public class JavaBook{
    Run | Debug
    public static void main(String []args){
        int [] num1={1,2,3,4,5};
        for(int anyname:num1){
            System.out.print("num1 : "+anyname+" , ");
            //output: num1 : 1
            //          num1 : 2
            //          num1 : 3
            //          num1 : 4
            //          num1 : 5
        }
        String [] name1={"Muhammad","Allah","Rakha"};
        for(String any:name1){
            System.out.print("name1 : "+any+" ");
            //output: name1 : Muhammad
            //          name1 : Allah
            //          name1 : Rakha
        }
    }
}
```

BOOK OF CODING

AAAASSTAAARK

```
import java.util.*;
public class JavaBook{
    Run | Debug
    public static void main(String []args){
        int a=10;
        if(a==10){
            System.out.print("1: is work");
        }
        else{
            System.out.print("2: is work");
        }
        //output: 1: is work

        if(a!=10){
            System.out.print("1: is work");
        }else if(a==10){
            System.out.println("2: is work");
        }else{
            System.out.print("3: is work");
        }
        //output: 2: is work

        String name2="aaaa";
        switch(name2){
            case "stark":
                System.out.print("1: is work");
                break;
            case "aaaastark":
                System.out.print("2: is work");
                break;
            case "aaaa":
                System.out.print("3: is work");
                break;
            default:
                System.out.println("Not work");
        }
        //output: 3: is work
    }
}
```

```
import java.util.*;
public class JavaBook{
    Run | Debug
    public static void main(String []args){
        Integer num1=5; Integer num2=6;
        System.out.print(num1.compareTo(6));
        //output: -1
        System.out.println(num1.equals(num2));
        //output: false
        Integer num3=Integer.valueOf(10);
        System.out.print(num3);
        //output: 10
        System.out.println(num3.toString());
        //output: 10 (this is a string of 10)
        System.out.println(Integer.toString(12));
        //output: 12 (this is a string of 12)
        Integer num4=-10;
        System.out.print(Math.abs(num4));
        //output: 10
        // here you can find Math.ceil(),Math.floor(),Math.round()
        // and Math.Log(),Math.pow( , ),Math.sqrt(),Math.sin(),Math.cos()
        // and Math.tan(),Math.random()
        System.out.println(Math.min(12,33));
        //output: 12
        System.out.println(Math.max(12,33));
        //output: 33
        //here you can use Character.toString() and Character.isUpperCase()
        // and Character.isLowerCase()
        System.out.print(Character.toUpperCase('c'));
        //output: C
        System.out.print(Character.toLowerCase('C'));
        //output: c
    }
}
```

BOOK OF CODING

AJAVA STARK

```
import java.util.*;
public class JavaBook{
    Run | Debug
    public static void main(String []args){
        String name1="M.A.R";
        System.out.print("Name is "+name1+"\n");
        //output: Name is M.A.R
        char[] name2={'M','.', 'A', '.', 'R'};
        String name3=new String(name2);
        System.out.print("Name : "+name3);
        //output: Name : M.A.R

        String names="Muhammad ";
        StringBuffer name4=new StringBuffer(names);
        name4.append("Allah Rakha");
        System.out.print(name4);
        //output: Muhammad ALLah Rakha
        name4.reverse();
        System.out.print(name4);
        //output: ahkaR halLA dammahuM
        name4.delete(0,6);
        System.out.print(name4);
        //output: halLA dammahuM
        name4.insert(0,"ahkaR ");
        System.out.println(name4);
        //output: ahkaR halLA dammahuM
        name4.replace(0,6,"Rakha");
        System.out.println(name4);
        //output: Rakha halLA dammahuM
    }
}

import java.util.*;
public class JavaBook{
    Run | Debug
    public static void main(String []args){
        String name1="AAAASTARK";
        char name2=name1.charAt(4);
        System.out.print(name2);
        //output: S
        int num1=name1.compareTo("aaaastark");
        System.out.print(num1);
        //output: -32

        String name3="Muhammad Allah Rakha ";
        name3=name3.concat("Naich");
        System.out.println(name3);
        //output: Muhammad ALLah Rakha Naich

        char[] name4={'M','.', 'A', '.', 'R'};
        String name5=" ";
        name5=name5.copyValueOf(name4);
        System.out.print(name5);
        //output: M.A.R
    }
}
```

BOOK OF CODING AAAAA STARK

```
import java.util.*;
public class JavaBook{
    Run | Debug
    public static void main(String []args){
        double[] name1={1.2,3.4,5.6,7};
        double total_num=0;
        double max_num=name1[0];
        for(int i=0;i<name1.length;i++){
            System.out.print(name1[i]+" ");
            //output: 1.2 3.4 5.6 7.0
        }
        for(int i=0;i<name1.length;i++){
            total_num+=name1[i];
        }
        for(int i=0;i<name1.length;i++){
            if(name1[i]>max_num){
                max_num=name1[i];
            }
        }
        System.out.println("Total Number : "+total_num);
        //output: Total Number : 17.2
        System.out.println("Max Number : "+max_num);
        //output: Max Number : 7.0
    }
}
```

```
// Advanced Java
import java.util.*;
public class JavaBook{
    int age;
    public JavaBook(String s){
        // this is constructor
        System.out.print("Name is "+s);
        //output: Name is AAAA STARK
    }
    public void set_age(int a){
        age=a;
    }
    public int get_age(){
        return age;
    }
}
Run | Debug
public static void main(String []args){
    JavaBook me =new JavaBook("AAAA STARK");
    me.set_age(20);
    Integer num1=me.get_age();
    System.out.print("Age is "+num1);
    //output: Age is 20
}
```

BOOK OF CODING

AAAAS TARK

```
import java.util.*;
public class JavaBook{
    String name1;
    Integer num1;
    Double num2;
    public JavaBook(String a,Integer b,Double c){
        this.name1=a; this.num1=b; this.num2=c;
        // you can use this Method e.g. (this.name1=a)
        // you can use this Method e.g. (name1=a)
    }
    public void sets_values(String d,Integer e,Double f){
        name1=d; num1=e; num2=f;
    }
    public void gets_values(){
        System.out.print("Name is "+name1+"\n");
        System.out.print("Age is "+num1+"\n");
        System.out.print("P_Number is "+num2);
    }
}
```

Run | Debug

```
public static void main(String []args){
    JavaBook anyname =new JavaBook("M.A.R",20,349.0175636);
    anyname.gets_values();
    //output: Name is M.A.R
    //          Age is 20
    //          P_Number is 349.0175636
    anyname.sets_values("AAAA STARK",21,309.0179917);
    anyname.gets_values();
    //output: Name is AAAA STARK
    //          Age is 21
    //          P_Number is 309.0179917
}
```

```
import java.util.*;
import java.io.*;
public class JavaBook{
    public String name1;
    private Integer age;
    public JavaBook(String s,Integer n){
        this.name1=s; this.age=n;
    }
    public void sets_values(String s,Integer n){
        name1=s; age=n;
    }
    public void gets_values(){
        System.out.print("Name is "+name1+"\n");
        //output: Name is M.A.R
        System.out.print("Age is "+age+"\n");
        //output: 20
    }
}
```

Run | Debug

```
public static void main(String []args){
    JavaBook anyname=new JavaBook("M.A.R",20);
    anyname.gets_values();
}
```

BOOK OF CODING



//Modifier Type

```
import java.util.*;
import java.io.*;
public class JavaBook{
    public String name;
    public void sets(String s){
        this.name=s;
    }
    public String gets(){
        return this.name;
    }
}
```

Run | Debug

```
public static void main(String []args){
    JavaBook anyname =new JavaBook();
    anyname.sets("MuhammadAllahRakha");
    String name1=anyname.gets();
    System.out.println("Name is "+name1);
    //output: Name is MuhammadAllahRakha
}
```

```
}
```

```
import java.util.*;
public class JavaBook{
// when you create (public static functions)
// then not allow (JavaBook anyname=new JavaBook());
    public static void name1(){
        System.out.println("Wellcome");
        //output: Wellcome
    }
    public static void name2(String n){
        System.out.println(" Mr. "+n);
        //output: Mr. AAAA STARK
    }
    public static String name3(){
        return "Hava a Nice day!";
    }
}
```

Run | Debug

```
public static void main(String[]args){
    name1();
    name2("AAAA STARK");
    String names=name3();
    System.out.print(names);
    //output: Hava a Nice day!
}
```

}

BOOK OF CODING

AAAASSTARTR

```
import java.util.*;
class Name1{
    String name;
}
class Name2{
    int age;
    String education;
    Name2(String n){
        this.education=n;
    }
}
public class JavaBook{
    Run | Debug
    public static void main(String[]args){
        Name1 name1=new Name1();
        Name2 name2=new Name2("BCS");
        name1.name="Muhammad Allah Rakha";
        name2.age=20;
        System.out.print("Name is "+name1.name);
        //output: Name is Muhammad Allah Rakha
        System.out.print("Age is "+name2.age);
        //output: Age is 20
        System.out.print("Education is "+name2.education);
        //output: Education is BCS
    }
}

import java.util.*;
public class JavaBook{
    public String name1;
    public void show(){
        System.out.print("AAAA STARK");
    }
    JavaBook(){
        System.out.print("Simple Constructer");
    }
    JavaBook(String n){
        this.name1=n;
        this.show();
    }
}
Run | Debug
public static void main(String[]args){
    JavaBook num1=new JavaBook();
    //output: Simple Constructer
    JavaBook num2=new JavaBook("M.A.R");
    //output: AAAA STARK
    System.out.print(num2.name1);
    //output: M.A.R
}
```

```
import java.util.*;
class My{
    private class Name1{
        private void prints(){
            System.out.print("AAAA STARK");
            //output: AAAA STARK
        }
    }
    // here create a function that access of private
    // class and data.Then you can easy in original class.
    void displys(){
        Name1 name1=new Name1();
        name1.prints();
    }
}
public class JavaBook{
    Run | Debug
    public static void main(String []args){
        My my=new My();
        my.displys();
    }
}
```

```
import java.util.*;
class My{
    public class Name1{
        public void prints(){
            System.out.print("AAAA STARK");
            //output: AAAA STARK
        }
    }
}
public class JavaBook{
    Run | Debug
    public static void main (String[] args){
        My my=new My();
        My.Name1 name1=my.new Name1();
        name1.prints();
    }
}
```

AAAAS TARK

```
import java.util.*;
public class JavaBook{
    void my_name(){
        System.out.print("AAAA STARK");
        //output: AAAA STARK
        class Me1{
            public void names(){
                System.out.print("M.A.R");
                //output: M.A.R
            }
        }
        Me1 me=new Me1();
        me.names();
    }
}
```

Run | Debug

```
public static void main (String[] args){
    JavaBook anyname=new JavaBook();
    anyname.my_name();
}
```

//Anonymous Class

```
import java.util.*;
abstract class Name{
    public abstract void name();
}
public class JavaBook{
    Run | Debug
    public static void main(String []args){
        Name num=new Name(){
            public void name(){
                System.out.print("M.A.R");
                //output: M.A.R
            }
        };
        num.name();
    }
}
```

BOOK OF CODING

```
// Java Object Oriented
//Java Inheritance

class Me{
    public void me1(){
        System.out.print("Muhammad Allah Rakha");
        //output: Muhammad Allah Rakha
    }
}

public class JavaBook extends Me{
    public void me2(){
        System.out.print("AAAA STARK");
        //output: AAAA STARK
    }

    Run | Debug
    public static void main(String []args){
        JavaBook anyname=new JavaBook();
        anyname.me1();
        anyname.me2();
    }
}

import java.util.*;
class Me{
    public void me1(){
        System.out.print("Muhammad Allah Rakha");
        //output: Muhammad Allah Rakha
    }

    public void me2(){
        System.out.print("Hassan Raza");
        //output: Hassan Raza
    }
}

public class JavaBook extends Me{
    public void me3(){
        System.out.print("AAAA STARK");
        //output: AAAA STARK
    }

    public void access_all(){
        JavaBook anynum=new JavaBook();
        anynum.me1();
        anynum.me2();
        anynum.me3();
    }

    // here you can easy eccess of upper class functions
}

Run | Debug
public static void main(String []args){
    JavaBook anyname=new JavaBook();
    // here you only one function access that access_of_all
    // class functions.
    anyname.access_all();
}
```

BOOK OF CODING

AAAAS TARK

```
import java.util.*;
class Me1{
    public void me1(){
        System.out.print("M.A.R");
    }
}
class Me2 extends Me1{
    public void me2(){
        System.out.print("M.A.R");
    }
}
class Me3 extends Me2{
    public void me3(){
        System.out.print("M.A.R");
    }
}
public class JavaBook extends Me3{
    Run | Debug
    public static void main(String []args){
        Me1 m1=new Me1();
        Me2 m2=new Me2();
        Me3 m3=new Me3();
        m1.me1(); //output: M.A.R
        m2.me2(); //output: M.A.R
        m3.me3(); //output: M.A.R
        JavaBook anyname=new JavaBook();
        anyname.me1(); //output: M.A.R
        anyname.me2(); //output: M.A.R
        anyname.me3(); //output: M.A.R
    }
}

import java.util.*;
class Me{
    public void me(){
        System.out.print("M.A.R");
    }
}
class My extends Me{
    public void me(){
        System.out.print("AAAA STARK");
    }
}
public class JavaBook extends My{
    Run | Debug
    public static void main(String []args){
        Me m1=new Me();
        My m2=new My();
        JavaBook anyname=new JavaBook();
        m1.me(); // M.A.R
        m2.me(); // AAAA STARK
        anyname.me(); // AAAA STARK
        // here you can access only one class (My)
    }
}
```

AAAASSTARK

BOOK OF CODING

```
//Encapsulation  
// here File name is Java1.java //  
public class Java1 {  
    private String name;  
    private int age;  
    public void set_age(int a){  
        this.age=a;  
    }  
    public void set_name(String s){  
        this.name=s;  
    }  
    // here you can use gets functions.  
    public void dis_values(){  
        System.out.print("Name is "+name);  
        System.out.print("Age is "+age);  
    }  
}  
// here File name is JavaBook.java //  
public class JavaBook{  
    Run | Debug  
    public static void main(String []args){  
        Java1 anyname=new Java1();  
        anyname.set_name("M.A.R");  
        anyname.set_age(20);  
        anyname.dis_values();  
        //output: Name is M.A.R  
        //output: Age is 20  
    }  
}
```

BOOK OF CODING

Java Star

```
//Polymorphism  
// here File name is Java1.java //  
public class Java1 {  
    private String name1;  
    public Java1(String name){  
        this.name1=name;  
    }  
    public void java1(){  
        System.out.print("Name "+name1);  
    }  
}  
// her file name is JavaBook.java //  
public class JavaBook{  
    Run | Debug  
    public static void main(String []args){  
        Java1 anyname=new Java1("M.A.R");  
        anyname.java1();  
        //output: Name is M.A.R  
    }  
}  
  
//Interfaces  
import java.lang.*;  
interface Me{  
    public void sets(String n);  
    public void gets();  
}  
public class JavaBook implements Me{  
    String name;  
    public void sets(String n){  
        this.name=n;  
    }  
    public void gets(){  
        System.out.print("Name is "+name);  
    }  
}  
Run | Debug  
public static void main(String []args){  
    JavaBook anyname=new JavaBook();  
    anyname.sets("M.A.R");  
    anyname.gets();  
    //output: Name is M.A.R  
}
```

C# (C-SHARP) PROGRAMMING LANGUAGE.

STRUCTURED, IMPERATIVE, OBJECT-ORIENTED, EVENT-DRIVEN, TASK-DRIVEN, FUNCTIONAL, GENERIC, REFLECTIVE, CONCURRENT

C# (pronounced see sharp, like the musical note C#, but written with the number sign) is a general-purpose, multi-paradigm programming language encompassing strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed around 2000 by Microsoft as part of its .NET initiative and later approved as an international standard by Ecma (ECMA-334) in 2002 and ISO (ISO/IEC 23270) in 2003. Mono is the name of the free and open-source project to develop a compiler and runtime for the language. C# is one of the programming languages designed for the Common Language Infrastructure (CLI).

CHAPEL, CLOJURE, CRYSTAL, D,
J#, DART, F#, HACK, JAVA,
KOTLIN, NEMERLE, OXYGENE,
RING, RUST, SWIFT, VALA, TYPE
SCRIPT

NAME

— IN THIS LANGUAGE —



AAAAA STARK
BOOK OF CODING

```
using System;

namespace CSharp_V.S.C
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("AAAA STARK");
            String name1="AAAA";
            string name2="STARK";
            Console.Write($"Name : {name1} {name2}");
            //output: Name : AAAA STARK
        }
    }
}
```

C#

— PROGRAMMING —

```
using System;

namespace CSharp_V.S.C
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.Write("Enter your name!");
            String name;
            name=Console.ReadLine();
            Console.Write("Enter your age!");
            int age;
            age=Console.Read();
            Console.Write($"Name :{name} Age :{age}");
            //output: Name :AAAA STARK Age :50
        }
    }
}
```



AAAASSТАРК

BOOK OF CODING

```
using System;

namespace CSharp_V.S.C
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            int num1=10;
            Console.WriteLine("Number "+num1);
            //output: Number 10
            double num2=12.34D; // D is must.
            Console.WriteLine("Number : "+num2);
            //output: Number : 12.34
            float num3=12.345F;
            Console.Write("Number is "+num3);
            //output: Number is 12.345
            String name1="Muhammad Allah Rakha";
            Console.Write("Name is "+name1);
            //output: Name is Muhammad Allah Rakha
            char name2='A';
            Console.Write("Name : "+name2);
            //output: Name : A
            bool name3=true;
            Console.Write("Name a "+name3);
            //output: Name a true (1)
        }
    }
}
```

```
using System;

namespace CSharp_V.S.C
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            int a=10;
            // here you can use (>,<,>=,<=, ==, !=)
            Console.Write(a>15);
            //output: False
            bool x=true;
            bool y=false;
            Console.Write(x&&y);
            //output: FalseFalse
            Console.Write(x||y);
            //output: True
            Console.Write(x!=y);
            //output: True
            int b=10;
            Console.Write(b==10?true:false);
            //output: True
            Console.Write(sizeof(int));
            //output: 4
            // here you use (double,short,int,float,long,etc.)
            var name1=19;
            Console.Write(++name1);
            //output: 20
            Console.WriteLine((name1++)+" "+name1);
            //output: 20      21
            // here you can use (--,++,+=, -=)
        }
    }
}
```



BOOK OF CODING

A AAAAASS TARK

```
using System;

namespace CSharp_V.S.C
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            int a=10;
            if(a>12){
                Console.WriteLine("1: work");
            }else{
                Console.WriteLine("2: work");
            }
            //output: 2: work
            if(a!=10){
                Console.WriteLine("1: work");
            }
            else if(a==10){
                Console.WriteLine("2: work");
            }
            else{
                Console.WriteLine("3: work");
            }
            //output: 2: work
        }
    }
}
```

A ~~AA~~AS TARK *BOOK OF CODING*

AAAASSТАРК

BOOK OF CODING

```
using System;

namespace CSharp_V.S.C
{
    0 references
    class Program
    {
        0 references
        public static void Main(string[] args)
        {
            int num=0;
            names(num);

//output: num : 0  num : 1  num : 2
//          num : 3  num : 4  num : 5
            Console.WriteLine("\n");
            names1();

//output: num1 : 0 num1 : 1 num1 : 2
//          num1 : 3 num1 : 4 num1 : 5

        }
        1 reference
        public static void names(int s){
            while(s<=5){
                Console.WriteLine($"num : {s}  ");
                s++;
            }
        }

        1 reference
        public static void names1(){
            for(int i=0;i<=5;i++){
                Console.WriteLine($"num1 : {i}  ");
            }
        }
    }
}
```

```
using System;

namespace CSharp_V.S.C
{
    0 references
    class Program
    {
        0 references
        public static void Main(string[] args)
        {
            names();
            names1(0);
        }

        1 reference
        public static void names(){
            string [] name={"Muhammad", "Allah", "Rakha"};
            foreach(string name1 in name){
                Console.Write($"{name1} ");
            }
            //output: Muhammad Allah Rakha
        }

        1 reference
        public static void names1(int i){
            do{
                Console.Write($"num2: {i} ");
                i++;
            }while(i<=5);
            //output: num2: 0  num2: 1  num2: 2
            //          num2: 3  num2: 4  num2: 5
        }
    }
}
```

```
using System;
```

```
namespace CSharp_V.S.C
```

```
{
```

```
    0 references
```

```
    class Program
```

```
{
```

```
    0 references
```

```
        public static void Main(string[] args)
```

```
{
```

```
            names();
```

```
}
```

```
    1 reference
```

```
    public static void names(){
```

```
        for(int i=0;i<4;i++){
```

```
            for(int j=0;j<4;j++){
```

```
                Console.Write(" M.A.R ");
```

```
}
```

```
}
```

```
//output: M.A.R  M.A.R  M.A.R  M.A.R
```

```
//          M.A.R  M.A.R  M.A.R  M.A.R
```

```
//          M.A.R  M.A.R  M.A.R  M.A.R
```

```
//          M.A.R  M.A.R  M.A.R  M.A.R
```

```
}
```

```
}
```

```
}
```

AAAAA STARK

BOOK OF CODING

```
using System;

namespace CSharp_V.S.C
{
    0 references
    class Program
    {
        0 references
        public static void Main(string[] args)
        {
            names();
            //output:Name : AAAA STARK
            Console.WriteLine("Age :" +ages());
            //output: Age : 20
            gets("FAST");
            //output: Study : FAST
        }

        1 reference
        public static void names(){
            Console.WriteLine("Name: AAAA STARK");
        }

        1 reference
        public static int ages(){
            return 20;
        }

        1 reference
        public static void gets(String s){
            Console.WriteLine("Study : " +s);
        }
    }
}
```

AAAAA STARK

BOOK OF CODING

```
using System;

namespace CSharp_V.S.C
{
    0 references
    class Program
    {
        0 references
        public static void Main(string[] args)
        {
            string name="M.A.R";
            names(ref name);
            Console.Write("Name2: "+name);
            //output: Name2: AAAA STARK
            int age;
            ages(out age);
            Console.Write("Age2 : "+age);
            //output: Age2 : 20
        }
        1 reference
        public static void names(ref string s){
            s="AAAA STARK";
            Console.Write($"Name1: {s}");
            //output: Name1: AAAA STARK
        }
        1 reference
        public static void ages(out int a){
            a=20;
            Console.Write("Age1 : "+a);
            //output: Age1 : 20
        }
    }
}
```

```
using System;

namespace CSharp_V.S.C
{
    0 references
    class Program
    {
        0 references
        public static void Main(string[] args)
        {
            int [] num=new int[10];
            names(num,10); // here sets of array(num)
            Console.Write("\n");
            names1(num,10); // here gets of array(num)
            Console.Write("\n");
            names2(num,10);
            Console.Write("\n");
            Console.Write(num);
            //output: System.Int32[]
        }

        1 reference
        public static void names(int[] array,int length){
            for(int i=0;i<length;i++){
                array[i]=i;
            }
            // here you can enter data of array from console.
        }

        1 reference
        public static void names1(int[] array1,int length1){
            for(int i=0;i<length1;i++){
                Console.Write($"Num : {array1[i]} ");
            }
        }

        //output: Num : 0 Num : 1 Num : 2 Num : 3 Num : 4
        //          Num : 5 Num : 6 Num : 7 Num : 8 Num : 9
    }

    1 reference
    public static void names2(int[] array2,int length2){
        int sum=0;
        for(int i=0;i<length2;i++){
            sum+=array2[i];
        }
        Console.Write("Sum of number : "+sum);
        //output: Sum of number : 45
    }
}
```

```
using System;

namespace CSharp_V.S.C
{
    0 references
    class Program
    {
        0 references
        public static void Main(string[] args)
        {
            names();
        }

        1 reference
        public static void names(){
            int [] array={1,2,3,4,5};
            Array.Reverse(array);
            foreach(int a1 in array){
                Console.Write($" {a1} ");
            }
            //output: 5 4 3 2 1
            Console.WriteLine();
            Array.Sort(array);
            foreach(int a1 in array){
                Console.Write($" {a1} ");
            }
            //output: 1 2 3 4 5
        }
    }
}
```

```
using System;

namespace CSharp_V.S.C
{
    0 references
    class Program
    {
        0 references
        public static void Main(string[] args)
        {
            int [] array={1,2,3,4,5};
            int [] num=new int[5];
            num=names(array);
            // here return array from function
            names1(num);

        }

        1 reference
        public static int[] names(int [] array){
            Array.Reverse(array);
            return array;
        }

        1 reference
        public static void names1(int [] arrays){
            foreach(int a1 in arrays){
                Console.Write($" {a1} ");
            }
            //output: 5 4 3 2 1
        }

    }
}
```

AAAASSTAAARK

using System;

```
namespace CSharp_V.S.C
```

```
{
```

```
    4 references
```

```
    class Program
```

```
{
```

```
    2 references
```

```
        private string name;
```

```
    2 references
```

```
        private int age;
```

```
    1 reference
```

```
        public Program(){
```

```
// here default constructer call
```

```
// therefor ever things is zero
```

```
}
```

```
    1 reference
```

```
        public Program(string s,int a){
```

```
            name=s; age=a;
```

```
// here parameter constructer call
```

```
// therefor ever things is set by values.
```

```
}
```

```
    2 references
```

```
        public void gets(){
```

```
            Console.WriteLine("Name is "+name);
```

```
            Console.WriteLine("Age is "+age);
```

```
}
```

```
    0 references
```

```
    public static void Main(string[] args)
```

```
{
```

```
        Program anyname=new Program();
```

```
        anyname.gets();
```

```
        //output: Name is Age is 0
```

```
        Program anyname1=new Program("M.A.R",20);
```

```
        anyname1.gets();
```

```
        //output: Name is M.A.RAge is 20
```

```
}
```

```
}
```

AUTHOR

MUHAMMAD ALLAH RAKHA

PROGRAMMING IN 15
LANGUAGE

AAAAA STARK SAY TO
YOU

I HOPE THIS BOOK IS PROVIDE
A SOMETHING KNOWLEDGE TO
YOU.I HOPE YOUR MIND IS
KNOW OF ALL POINT THATS
MENTION IN THIS BOOK AND
HOPE ARE YOU WELL.

MUHAMMAD ALLAH RAKHA

CONNECT

Phone: +92 3490175636

Email: 4444stark@gmail.com

LinkedIn: AAAA STARK

WhatsApp : +92 3490175636



STEP BY STEP
PROGRAMMING
IN 15 LANGUAGE

THE 2020 EDITION OF
AAAA STARK COMPANY



26/9/20, 12.00 AM, FROM PESHAWAR IN
PAKISTAN

الله اکبر محمد رسول اللہ