

P1: write a program to print Hello world

```
//Hello World Program

class Hello

{

public static void main(String []args)

{

System.out.println("Hello World");

}

}
```

P2: Write a program that prints the numbers from 1 to 100 and for multiples of '3' print "Fizz" instead of the number and for the multiples of '5' print "Buzz".

```
// Java program to print Fizz Buzz

import java.util.*;

class FizzBuzz

{

    public static void main(String[] args)

    {

        int n = 100;

        // loop for 100 times

        for (int i=1; i<=n; i++)

        {

            //number divisible by 15(divisible by

            // both 3 & 5), print 'FizzBuzz' in

            // place of the number

            if (i%15==0)

                System.out.println("FizzBuzz");

            // number divisible by 5, print 'Buzz'
```

```
        // in place of the number
        else if (i%5==0)
            System.out.println("Buzz");

        // number divisible by 3, print 'Fizz'
        // in place of the number
        else if (i%3==0)
            System.out.println("Fizz");

        else // print the numbers
            System.out.println(i);
    }
}
}
```

P3 : Program to tell whether the given number is even or odd

// Java Program to Check if Given Integer is Odd or Even

// Importing required classes

import java.io.*;

import java.util.Scanner;

class EvenOdd {

// Main Driver Method

public static void main(String[] args)

{ Scanner sc = new Scanner(System.in);

```
System.out.println("Enter any number to check even or odd:");

// Declaring and initializing integer variable
int num = sc.nextInt();

// Checking if number is even or odd number
// via remainder
if (num % 2 == 0) {

    // If remainder is zero then this number is even
    System.out.println("Entered Number is Even"+num);
}

else {

    // If remainder is not zero then this number is
    // odd
    System.out.println("Entered Number is Odd"+num);
}
}
```

P4: Program on Age Calculator

```
//Program of calculate age
import java.time.*;
import java.util.*;
class CalculateAge
{
public static void main(String args[])
{

```

```
System.out.print("Enter date of birth in YYYY-MM-DD format: ");

Scanner scanner = new Scanner(System.in);

//reads the date of birth from the user
String input = scanner.nextLine();
scanner.close();

//obtains an instance of LocalDate from a year, month and date
LocalDate dob = LocalDate.parse(input);

//obtains the current date from the system clock
LocalDate curDate = LocalDate.now();

//calculates the difference between two dates
Period period = Period.between(dob, curDate);

//prints the difference in years, months, and days
System.out.printf("Your age is %d years %d months and %d days.", period.getYears(),
period.getMonths(), period.getDays());
}
}
```

P5: Program on Temperature Conversion

```
//Program to convert temperature from celsius to fahrenheit
class temperature
{
    public static void main (String args[])
    { float Fahrenheit, Celsius;
        Celsius= 13;
        Fahrenheit=((Celsius*9)/5)+32;
        System.out.println("Temperature in Fahrenheit is: "+Fahrenheit);
    }
}
```

```
}}
```

P6: Weight on Mars calculator is one of the fun and simple programs of Java to implement as a beginner. This program calculates a person's weight on Mars based on their weight on Earth. All you need to do is ask the user for their weight on Earth and then print out their weight on Mars. In order to convert weight on Earth to weight on Mars, use the coefficient of 0.38. This means that if someone weighs 100 pounds on Earth, they would weigh 38 pounds on Mars.

```
//Program to calculate weight on Mars
```

```
import java.util.Scanner;
```

```
class SpaceBoxing
```

```
{
```

```
    static Scanner keyboard = new Scanner(System.in);
```

```
    public static void main(String[] args)
```

```
    {
```

```
        double weight = getEarthWeight();
```

```
        int choice = getPlanetMars();
```

```
        double realWeight = computeEffectiveWeight(weight, choice);
```

```
        System.out.println("Your weight would be " + realWeight + " on Mars planet. ");
```

```
    }
```

```
    static double getEarthWeight()
```

```
    {
```

```
        System.out.println("Please enter your current Earth weight: ");
```

```
        return keyboard.nextDouble();
```

```
    }
```

```
    static int getPlanetMars()
```

```
{  
    System.out.println("I have information of Mars planet below: ");  
    System.out.println("1. Mars");  
    return keyboard.nextInt();  
}
```

```
static double computeEffectiveWeight(double weight, int choice)  
{  
    double realWeight = 0;  
    if(choice == 1)  
        realWeight = (weight * 0.38);  
  
    return realWeight;  
}  
}
```

2nd Solution

```
/*  
import java.util.Scanner;  
  
public class SpaceBoxing  
{  
    static Scanner keyboard = new Scanner(System.in);  
  
    public static void main(String[] args)  
    {  
        double weight = getEarthWeight();  
        int choice = getPlanetChoice();  
        double realWeight = computeEffectiveWeight(weight, choice);  
        System.out.println("Your weight would be " + realWeight + " on that planet. ");  
    }  
}
```

```
}
```

```
static double getEarthWeight()
```

```
{
```

```
    System.out.println("Please enter your current Earth weight: ");
```

```
    return keyboard.nextDouble();
```

```
}
```

```
static int getPlanetChoice()
```

```
{
```

```
    System.out.println("I have information on the following planets: ");
```

```
    System.out.println("1. Venus");
```

```
    System.out.println("2. Mars");
```

```
    System.out.println("3. Jupiter");
```

```
    System.out.println("4. Saturn");
```

```
    System.out.println("5. Uranus");
```

```
    System.out.println("6. Neptune");
```

```
    return keyboard.nextInt();
```

```
}
```

```
static double computeEffectiveWeight(double weight, int choice)
```

```
{
```

```
    double realWeight = 0;
```

```
    if(choice == 1)
```

```
        realWeight = (weight * 0.78);
```

```
    else if(choice == 2)
```

```
        realWeight = (weight * 0.39);
```

```
        else if(choice == 3)
            realWeight = (weight * 2.65);

        else if(choice == 4)
            realWeight = (weight * 1.17);

        else if(choice == 5)
            realWeight = (weight * 1.05);

        else if(choice == 6)
            realWeight = (weight * 1.23);

        return realWeight;
    }
}

*/
```

P7 : This program calculates the distance between two cities based on their latitude and longitude coordinates. All you need to do is ask the user for the latitude and longitude of two cities and then print out the distance between them.

```
// Java program to calculate Distance Between
// Two Points on Earth
import java.util.*;
import java.lang.*;

class Distance {

    public static double distance(double lat1,
```



```
        double lat2, double lon1,
            double lon2)
{

    // The math module contains a function
    // named toRadians which converts from
    // degrees to radians.
    lon1 = Math.toRadians(lon1);
    lon2 = Math.toRadians(lon2);
    lat1 = Math.toRadians(lat1);
    lat2 = Math.toRadians(lat2);

    // Haversine formula
    double dlon = lon2 - lon1;
    double dlat = lat2 - lat1;
    double a = Math.pow(Math.sin(dlat / 2), 2)
        + Math.cos(lat1) * Math.cos(lat2)
        * Math.pow(Math.sin(dlon / 2), 2);

    double c = 2 * Math.asin(Math.sqrt(a));

    // Radius of earth in kilometers. Use 3956
    // for miles
    double r = 6371;

    // calculate the result
    return(c * r);
}
```

```
// driver code

public static void main(String[] args)
{
    double lat1 = 53.32055555555556;
    double lat2 = 53.31861111111111;
    double lon1 = -1.7297222222222221;
    double lon2 = -1.6997222222222223;
    System.out.println(distance(lat1, lat2,
                                lon1, lon2) + " K.M");
}
}
```

P8 This program calculates the amount of interest earned on an investment over a period of time and is among simpler programs of Java for a starting Java developer. All you need to do is ask the user for the initial investment amount, the interest rate, and the number of years. Then print out the total amount of interest earned. Use Java exponentiation to calculate the compounding interest.

```
//Java Program to calculate the compound interest

import java.util.*;

class CompoundInterest
{
    public static void main(String args[])
    {
        //Take input from the user
        //Create an instance of the Scanner class
        Scanner sc = new Scanner(System.in);

        //Declare variables
        float p, r, t, n;

        System.out.println("Enter the Principal : ");

        p = sc.nextFloat(); //Initialize the variables
```

```
System.out.println("Enter the Rate of interest : ");

r = sc.nextFloat(); //Initialize the variables

System.out.println("Enter the Time period : ");

t = sc.nextFloat(); //Initialize the variables

System.out.println("Enter the number of times that interest is compounded per unit t");

n=sc.nextFloat(); //Initialize the variables

sc.close();


//Calculate the compound interest

double amount = p * Math.pow(1 + (r / n), n * t);

double cinterest = amount - p;

System.out.println("Compound Interest after " + t + " years: "+cinterest);

System.out.println("Amount after " + t + " years: "+amount);

}

}
```

P9: This program should print a diamond shape to the console. The user should specify the height of the diamond, which will always be an odd number. For example, if the user specifies a height of 5.

```
// JAVA Code to print
// the diamond shape
import java.util.*;

class Diamond
{

    // Prints diamond pattern
    // with 2n rows
    static void printDiamond(int n)
    {
```

```
int space = n - 1;

// run loop (parent loop)
// till number of rows
for (int i = 0; i < n; i++)
{
    // loop for initially space,
    // before star printing
    for (int j = 0; j < space; j++)
        System.out.print(" ");

    // Print i+1 stars
    for (int j = 0; j <= i; j++)
        System.out.print("* ");

    System.out.print("\n");
    space--;
}

// Repeat again in
// reverse order
space = 0;

// run loop (parent loop)
// till number of rows
for (int i = n; i > 0; i--)
{
    // loop for initially space,
```

```
// before star printing
for (int j = 0; j < space; j++)
    System.out.print(" ");

// Print i stars
for (int j = 0; j < i; j++)
    System.out.print("* ");

System.out.print("\n");
space++ ;
    }
}

// Driver Code
public static void main(String[] args)
{
    printDiamond(5);
}
}
```

P10: Given an array of 0s 1s and 2s .Segregate them

```
import java.util.Arrays;
```

```
class Main {
    public static void segregate(int arr[], int n) {
        int count_0 = 0, count_1 = 0, count_2 = 0;
```

// Step 1: Count the number of 0's, 1's, and 2's

```
for (int i = 0; i < n; i++) {  
    if (arr[i] == 0) {  
        count_0++;  
    } else if (arr[i] == 1) {  
        count_1++;  
    } else if (arr[i] == 2) {  
        count_2++;  
    }  
}
```

// Step 2: Overwrite the array with the correct number of each value

```
int i = 0;  
while (count_0 > 0) {  
    arr[i] = 0;  
    count_0--;  
    i++;  
}  
while (count_1 > 0) {  
    arr[i] = 1;  
    count_1--;  
    i++;  
}  
while (count_2 > 0) {  
    arr[i] = 2;  
    count_2--;  
    i++;  
}  
}
```

```
public static void main(String[] args) {  
    int arr[] = {2, 0, 1, 2, 0, 1, 0, 2, 1};  
    int n = arr.length;  
  
    System.out.print("Array: ");  
    for (int i = 0; i < n; i++) {  
        System.out.print(arr[i] + " ");  
    }  
    System.out.println();  
  
    segregate(arr, n);  
  
    System.out.print("Segregated array: ");  
    for (int i = 0; i < n; i++) {  
        System.out.print(arr[i] + " ");  
    }  
    System.out.println();  
}  
}
```

P11: Program to find the sum of all elements in the array.

```
class TestArray {  
  
    public static void main(String[] args) {  
        double[] myList = {1.9, 2.9, 3.4, 3.5};  
  
        // Print all the array elements  
        for (int i = 0; i < myList.length; i++) {  
            System.out.println(myList[i] + " ");  
        }  
    }  
}
```

```
// Summing all elements
double total = 0;
for (int i = 0; i < myList.length; i++) {
    total += myList[i];
}
System.out.println("Total is " + total);
```

```
// Finding the largest element
//double max = myList[0];
//for (int i = 1; i < myList.length; i++) {
//    // if (myList[i] > max) max = myList[i];
//}
//System.out.println("Max is " + max);
}
}
```

P12 Write a Java Program to Create the Object and the use of it.

```
public class Car
{
    private String car_color;
    private String brand;
    private String fuel_type;
    public void Start()
    {
        System.out.println("Car Started");
    }
    public void Stop()
    {
        System.out.println("Car Stopped");
    }
}
```



```
public static void main(String args[])
{
    //creating an object of Car class
    Car car = new Car();
    //Calling method Start() using object
    car.Start();
    //Calling method Stop() using object
    car.Stop();
}
}
```

P13 This program should generate a random number between 1 and 100, and then ask the user to guess the number. If the user's guess is too high or too low, the program should give them a hint and let them try again. The program should keep track of how many guesses the user has made, and end when the user has guessed the number correctly.

```
// import required classes for the program
import java.util.Scanner;
import java.lang.Math;

class GuessingGame
{
    public static void main(String[] args)
    {
        // generate a random number between 1 and 100
        int answer = (int)(Math.random() * 100) + 1;
        // number of trials that the user has to guess the number
        int k = 5;
        // create a scanner object to read user input
        Scanner input = new Scanner(System.in);
        // TO check if the user has guessed the number
        boolean correct = false;
        System.out.println("I'm thinking of a number between 1 and 100.\nYou have 5 tries to guess the number.");
        while (k > 0) {
            System.out.println("Enter your guess: ");
            int guess = input.nextInt();
            // if the user guesses correctly, print the congratulation message and exit the program
            if (guess == answer) {
                System.out.println("You guessed the number!\nYou win!");
                break;
            }
            // if the user guesses greater than the number, print the message and reduce the number of
            // trials
            else if (guess > answer) {
                System.out.println("Your guess is too high.\nYou have " + (k - 1) + " tries left.");
                // k--;
            }
        }
    }
}
```

```

// if the user guesses less than the number, print the message and reduce the number of
// trials
else {
    System.out.println("Your guess is too low.\nYou have " + (k - 1) + " tries left.");
}
// after each trial decrease the number of trials by 1
k--;
}
// if the user has run out of trials, print the message and exit the program

if (correct == false) {
    System.out.println("You ran out of tries.\nYou lose!");
}
System.out.println("Correct number is:"+answer);
}
}

```

P14. a) Program to show the importance of Encapsulation

```

class Computer
{
    private String motherboard;
    private String ramsize;

    public void setmotherboard(String motherboard)
    {
        this.motherboard = motherboard;
    }
    public void setramsize(String ramsize)
    {
        this.ramsize = ramsize;
    }
    public String getramsize()
    {
        System.out.println("Ramsize is :"+ramsize);
        return ramsize;
    }
    public String getmotherboard()
    {
        System.out.println("motherboard is"+motherboard);
        return motherboard;
    }
}
class Main
{
    public static void main(String []args)
    {
        Computer c1 = new Computer();
        c1.setmotherboard("sps");
        c1.setramsize("16gb");
    }
}

```

P14. b) Program to show the importance of Inheritance

```
public class Parent
{
    int z;
    public void add(int x ,int y)
    {
        z = x+y;
        System.out.println(z);
        //System.out.println("Addition of two numbers is:",z);

    }
    public void sub(int x ,int y)
    {
        z = x-y;
        System.out.println("Addition of two numbers is:",z);

    }

}

class Child extends Parent
{
    public void mul(int x,int y)
    {
        z= x*y;
        System.out.println("Addition of two numbers is:",z);

    }
    public static void main(String[] args) {
        Child c1 = new Child();
        c1.mul(10,20);
        c1.add(10,20);
        c1.sub(10,20);
    }

}
```

P14. c) Program to show the Importance of Polymorphism

// Java Program for Method overloading
// By using Different Types of Arguments

Class 2

Main class

```
class Main {

    // Main driver method
    public static void main(String[] args)
    {

        // Calling method by passing
        // input as in arguments
        System.out.println(Helper.Multiply(2, 4));
        System.out.println(Helper.Multiply(5.5, 6.3));

    }

}
```

```
// Class 1
// Helper class
class Helper {

    // Method with 2 integer parameters
    static int Multiply(int a, int b)
    {

        // Returns product of integer numbers
        return a * b;
    }

    // Method 2
    // With same name but with 2 double parameters
    static double Multiply(double a, double b)
    {

        // Returns product of double numbers
        return a * b;
    }
}
```

// Java Program for Method Overriding

```
// Class 4
// Main class
class Main {

    // Main driver method
    public static void main(String[] args)
    {

        // Creating object of class 1
        Parent a;

        // Now we will be calling print methods
        // inside main() method

        a = new subclass1();
        a.Print();

        a = new subclass2();
        a.Print();
    }
}

// Class 1
// Helper class
class Parent {

    // Method of parent class
    void Print()
    {
```

```
        // Print statement
        System.out.println("parent class");
    }
}

// Class 2
// Helper class
class subclass1 extends Parent {

    // Method
    void Print() { System.out.println("subclass1"); }
}

// Class 3
// Helper class
class subclass2 extends Parent {

    // Method
    void Print()
    {

        // Print statement
        System.out.println("subclass2");
    }
}
```

P14.d) d) Program to use super with variables

```
class Main
{
    public static void main(String args[]) {
        Sub_class obj = new Sub_class();
        obj.my_method();
    }
}

class Super_class {
    int num = 20;

}

public class Sub_class extends Super_class {
    int num = 10;

    public void my_method() {
        // Instantiating subclass
        Sub_class sub = new Sub_class();

        // printing the value of variable num of subclass
        System.out.println("value of the variable named num in sub class:"+ sub.num);
    }
}
```

```
        // printing the value of variable num of superclass
        System.out.println("value of the variable named num in super class:"+ super.num);
    }
}
```

P15: Program to use super with methods

```
class Main
{
    public static void main(String args[]) {
        Sub_class obj = new Sub_class();
        obj.my_method();
    }
}

class Super_class {

    // display method of superclass
    public void display() {
        System.out.println("This is the display method of superclass");
    }
}

public class Sub_class extends Super_class {

    // display method of sub class
    public void display() {
        System.out.println("This is the display method of subclass");
    }

    public void my_method() {
        // Instantiating subclass
        Sub_class sub = new Sub_class();

        // Invoking the display() method of sub class
        sub.display();

        // Invoking the display() method of superclass
        super.display();

    }
}
```

P16: Program to find the Importance and working of Inner class

```
class Main
{
    public static void main (String []args){
        Outer o = new Outer ();
        Outer.Inner i = o.new Inner();
        System.out.println(o.x);
        System.out.println(i.y);
    }
}
```

```
}  
class Outer  
{  
    int x = 10;  
    class Inner  
    {  
        int y=5;  
    }  
}
```

P17: Program on Implementing Interfaces

```
class Main  
{  
    public static void main (String []args)  
    {  
        Bike h = new Honda();  
        h.run();  
        Bike t = new Tvs();  
        t.run();  
    }  
}  
interface Bike  
{  
    void run();  
}  
class Honda implements Bike  
{  
    public void run()  
    {  
        System.out.println("Honda speed");  
    }  
}  
class Tvs implements Bike  
{  
    public void run()  
    {  
        System.out.println("Tvs speed");  
    }  
}
```

P18: Program on Creating Packages

```
package myPackage;  
  
public class MyClass  
{  
    public void getNames(String s)  
    {  
        System.out.println(s);  
    }  
}  
  
import myPackage.MyClass;  
  
public class PrintName
```

```
{
public static void main(String args[])
{
    // Initializing the String variable
    // with a value
    String name = "Parul University";

    // Creating an instance of class MyClass in
    // the package.
    MyClass obj = new MyClass();

    obj.getNames(name);
}
}
```

// Note : MyClass.java must be saved inside the myPackage directory since it is a part of the package.

P19: a) Program on String Buffer and its methods

**// Java program to demonstrate
// the StringBuffer class**

```
public class stringbuffer {

    // Driver code
    public static void main(String args[])
    {

        // Creating a new StringBuffer
        StringBuffer str = new StringBuffer("Hello");

        str.append(" World!");

        System.out.println(str);
    }
}
```

b) Program on String Builder and String Buffer methods

**// Java program to demonstrate
// the StringBuilder class**

```
public class stringbuilder {

    // Driver code
    public static void main(String args[])
    {

        // Creating a new StringBuilder
        StringBuilder str
            = new StringBuilder("Hello");

        str.append(" World!");

        System.out.println(str);
    }
}
```



```
    }
}
```

c) Program on String Manipulation

```
public class Main{

    public static void main(String args[]){
        char[] c={'j','a','v','a'};
        String s1=new String(c);

        String s2="java";

        System.out.println(s1);
        System.out.println(s2);
        System.out.println(s1.length());
        System.out.println(s2);
        System.out.println(s2.concat(s1));
        System.out.println(s1.substring(1));
        System.out.println(s1.substring(1,2));
        System.out.println(s1.replace('a','e'));
        System.out.println(s1.replace("in","within"));
        System.out.println(s1.trim());
        System.out.println(s1.toUpperCase());
        System.out.println(s1.toLowerCase());
        System.out.println(s1.equals(s2));
    }
}
```

d) Programs of enum

```
class EnumExample1{
//defining the enum inside the class
public enum Season { WINTER, SPRING, SUMMER, FALL }
//main method
public static void main(String[] args) {
//traversing the enum
for (Season s : Season.values())
System.out.println(s);
}}
```

P20.Program on Exceptional Handling

try catch program

```
public class JavaExceptionExample{
    public static void main(String args[]){
        try{
            //code that may raise exception
            int data=100/0;
        }catch(ArithmeticException e){System.out.println(e);}
        //rest code of the program
        System.out.println("rest of the code...");
    }
}
```

finally program

```
class TestFinallyBlock {
    public static void main(String args[]){
        try{
            //below code do not throw any exception
            int data=25/5;
            System.out.println(data);
        }
        //catch won't be executed
        catch(NullPointerException e){
            System.out.println(e);
        }
        //executed regardless of exception occurred or not
        finally {
            System.out.println("finally block is always executed");
        }

        System.out.println("rest of the code...");
    }
}
```

throw keyword program**TestThrow.java**

```
public class TestThrow {
    //defining a method
    public static void checkNum(int num) {
        if (num < 1) {
            throw new ArithmeticException("\nNumber is negative, cannot calculate square");
        }
        else {
            System.out.println("Square of " + num + " is " + (num*num));
        }
    }
    //main method
    public static void main(String[] args) {
        TestThrow obj = new TestThrow();
        obj.checkNum(-3);
        System.out.println("Rest of the code..");
    }
}
```

throws keyword program

```
public class TestThrows {
    //defining a method
    public static int divideNum(int m, int n) throws ArithmeticException {
        int div = m / n;
        return div;
    }
    //main method
```

```
public static void main(String[] args) {
    TestThrows obj = new TestThrows();
    try {
        System.out.println(obj.divideNum(45, 0));
    }
    catch (ArithmeticException e){
        System.out.println("\nNumber cannot be divided by 0");
    }

    System.out.println("Rest of the code..");
}
}
```

TestThrowAndThrows

```
public class TestThrowAndThrows
{
    // defining a user-defined method
    // which throws ArithmeticException
    static void method() throws ArithmeticException
    {
        System.out.println("Inside the method()");
        throw new ArithmeticException("throwing ArithmeticException");
    }
    //main method
    public static void main(String args[])
    {
        try
        {
            method();
        }
        catch(ArithmeticException e)
        {
            System.out.println("caught in main() method");
        }
    }
}
```