**Name :** RidaShabbir

**Student id =** BCSF20M541

**WEB\_01:**

**Question # 1 :**

import java.util.\*;

import java.io.\*;

class A {

private void main (String[] str){

System.println("Hello java");

}

}

//E rror # 1

//the main function should be declared public so that the program can access it as when a java program runs

//the compiler starts its execution from the main function so the main function + the class containing the main function should be declared public

//Error#2

//System.println is not a defined symbol in java

//Instead write System.out.println("Desired Message")

Question # 2 :

There are Two types of type casring in java

1. Upcasting, Widening (convert small data type into large data type, No imformation loss)
2. Downcasting, Narrowing (convert large data type to small data type, Some information loss )

**Upcasting example:**

import java.util.\*;

import java.io.\*;

class B {

public static void main (String[] str){

int i = 5;

float f;

long l;

f = i;

l = i;

System.out.println(i);

System.out.println(f);

System.out.println(l);

}

}

**Downcasting example:**

import java.util.\*;

import java.io.\*;

public class C {

public static void main (String[] str){

double d=8;

int i;

long l;

i = (int)d;

l = (long)d;

System.out.println(i);

System.out.println(d);

System.out.println(l);

}

}

**Question # 3 :**

import java.util.Scanner;

import java.io.\*;

public class D {

public static void main (String[] str){

Scanner scanner = new Scanner(System.in);

int min=99, max=0, sum=0, count=0;

int num=0;

while(true){

System.out.println("Enter a Number: ");

num = scanner.nextInt();

if(num == -999) break;

else {

if(num>max) max = num;

if(num<min) min = num;

sum = sum + num;

count++; }

}

System.out.println("Minimum Number: " + min );

System.out.println("Maximum Number: " + max );

System.out.println("Sum of Numbers: " + sum );

System.out.println("Average of Numbers: " + (sum/count) );

}

}

**Question # 4 :**

import java.util.Scanner;

import java.io.\*;

public class E {

public static void main (String[] str){

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a Number: ");

int num = scanner.nextInt();

if(num<0) System.out.println("It is a Neagtive Number.");

else System.out.println("It is a Positive Number.");

}

}

**Question # 5 :**

import java.util.Scanner;

import java.lang.Math;

import java.io.\*;

public class G {

public static void main (String[] str){

//Scanner scanner = new Scanner(System.in);

//System.out.println("Enter a Number: ");

//int num = scanner.nextInt();

for ( int num =0; num<=1000; num++) {

int[] a = new int[4];

int i=0, n=num, n1=num;

while (n1!=0) {

a[i] = n1%10;

n1 = n1/10;

i = i+1;

}

double sum =0;

for( int j=0; j<i; j++ ) {

int value = a[j];

sum = sum + Math.pow(value, i);

}

if(sum==n) System.out.println("Number is Armstrong "+ n);

else System.out.println("Number is Not Armstrong "+ n);

}

}

}

**WEB\_02:**

**Question # 1 :**

import java.util.Scanner;

import java.lang.Math;

import java.io.\*;

public class A{

public static void main (String[] str){

Scanner scanner = new Scanner(System.in);

System.out.println("Enter Number of rows and columns: ");

int rows = scanner.nextInt();

int cols = scanner.nextInt();

int[][] matrix = new int[rows][cols];

int[][] transpose\_matrix = new int[cols][rows];

System.out.println("Enter the Values of Matrix: ");

for ( int i=0; i<rows; i++ ){

for( int j=0; j<cols; j++ ){

matrix[i][j] = scanner.nextInt();

}

}

System.out.println("Transpose of matrix: ");

for (int i = 0; i < cols; i++) {

for (int j = 0; j < rows; j++) {

transpose\_matrix[i][j] = matrix[j][i];

System.out.print(transpose\_matrix[i][j] + " ");

}

System.out.println();

}

}

}

**Question # 2 :**

import java.util.Scanner;

import java.lang.Math;

import java.io.\*;

public class B{

public static void main (String[] arg){

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a Number: ");

int num = scanner.nextInt();

int flag = 1;

String str = Integer.toString(num);

int j = str.length() - 1;

for( int i=0; i<str.length()/2; i++ ) {

if(str.charAt(i)==str.charAt(j)) {

j--;

continue;

}

else {

flag = 0;

break;

}

}

if(flag==0) System.out.println("Number is not a Palindrom.");

else System.out.println("Number is a Palindrom.");

}

}

**Question # 03 :**

import java.util.Scanner;

import java.lang.Math;

import java.io.\*;

public class C{

public static void main (String[] arg){

Scanner scanner = new Scanner(System.in);

System.out.print("Input String: ");

String str = scanner.nextLine();

System.out.println("Result: ");

for( int i=0; i<str.length(); i++ ) {

if(str.charAt(i)==' ') System.out.print("\n");

else System.out.print(str.charAt(i));

}

System.out.print("\n");

}

}

**Question # 4 :**

interface StudentData {

public void setName(String n);

public void setRollNumber(int r);

public void setCGPA(double c);

public String getName();

public int getRollNumber();

public double getCGPA();

public void showData();

}

public class Student implements StudentData {

String name;

int RollNumber;

double CGPA;

public void setName( String name){

this.name = name;

}

public void setRollNumber( int RollNumber){

this.RollNumber = RollNumber;

}

public void setCGPA( double CGPA){

this.CGPA = CGPA;

}

public String getName(){

return name;

}

public int getRollNumber(){

return RollNumber;

}

public double getCGPA(){

return CGPA;

}

public void showData(){

System.out.println("Student Data: ");

System.out.println("Student Name: " + getName());

System.out.println("Student Roll Number: " + getRollNumber());

System.out.println("Student CGPA: " + getCGPA());

}

public static void main(String[] args) {

Student s = new Student();

s.setName("Rida");

s.setRollNumber(41);

s.setCGPA(3.3);

s.getName();

s.getRollNumber();

s.getCGPA();

s.showData();

}

}

**Question # 5 :**

class Shape{

String color;

boolean filled;

//Constructor

public Shape (){

this.color = "Red";

this.filled = true;

}

public Shape (String color, boolean filled){

this.color = color;

this.filled = filled;

}

public void setColor (String color){

this.color = color;

}

public String getColor () {

return color;

}

public void setFilled (boolean filled){

this.filled = filled;

}

public boolean isFilled () {

return filled;

}

public String toString() {

return "Shape: \nColor=" + color + "\nFilled =" + filled + "\n";

}

}

class Circle extends Shape{

double radius;

public Circle (){

this.radius = 1.0;

}

public Circle (double radius){

this.radius = radius;

}

public Circle ( String color, boolean filled, double radius ){

super(color,filled);

this.radius = radius;

}

public void setRadius ( double radius ) {

this.radius = radius;

}

public double getRadius () {

return radius;

}

public double getArea () {

return 3.14159\*radius\*radius;

}

public double getPerimeter () {

return 2\*3.14159\*radius;

}

public String toString() {

return "Circle: \n Radius= " + getRadius() + "\nArea =" + getArea() + "\nPerimeter =" + getPerimeter() +"\n";

}

}

class Rectangle extends Shape{

double width;

double length;

public Rectangle (){

this.width = 1.0;

this.length = 1.0;

}

public Rectangle (double width, double length){

this.width = width;

this.length = length;

}

public Rectangle ( String color, boolean filled, double width, double length ){

super(color,filled);

this.width = width;

this.length = length;

}

public void setWidth ( double width ) {

this.width = width;

}

public double getWidth () {

return width;

}

public void setLength ( double length ) {

this.length = length;

}

public double getLength () {

return length;

}

public double getArea () {

return length\*width;

}

public double getPerimeter () {

return (2\*length)+(2\*width);

}

public String toString() {

return "Rectangle: \nWidth= " + getWidth() + "\nLength =" + getLength() + "\nPerimeter =" + getPerimeter() + "\nArea =" + getArea() +"\n";

}

}

public class D {

public static void main ( String[] args ) {

Shape s = new Shape("Blue",true);

System.out.println("Shape Function: " + s.toString());

Circle c = new Circle("Blue",true, 1.5);

System.out.println("\nCircle Function: " + c.toString());

Rectangle r = new Rectangle("Blue",true, 1.5,1.5);

System.out.println("\nRectangle Function: " + r.toString());

}

}

**Question # 6 :**

class Book{

String name;

int ISBN;

String author;

String publisher;

public Book (String name, int ISBN, String author,String publisher){

this.name = name;

this.ISBN = ISBN;

this.author = author;

this.publisher = publisher;

}

public void setName (String name){

this.name = name;

}

public String getColor () {

return name;

}

public void setISBN (int ISBN){

this.ISBN = ISBN;

}

public int getISBN () {

return ISBN;

}

public void setAuthor (String author){

this.author = author;

}

public String getAuthor () {

return author;

}

public void setPublisher (String publisher){

this.publisher = publisher;

}

public String getPublisher () {

return publisher;

}

public String getBookInfo() {

return "Name= " + name + "\nISBN= " + ISBN + "\nAuthor= " + author + "\nPublisher= " + publisher + "\n";

}

}

class BookTest {

public static void main(String[] args){

Book test[] = new Book[13];

test[1] = new Book("Atomic Habits", 1234, "James Clear", "Avery Publishing Group");

System.out.println("Book 01 Information: \n" + test[1].getBookInfo() );

test[2] = new Book("Kaiser O Kusra", 777, "Naseem hijazi", "Urdu Bazar");

System.out.println("\nBook 02 Information: \n" + test[2].getBookInfo() );

}

}

**Bonus Task:**

**import java.util.Scanner;**

**public class TicTacToe {**

**private static final char EMPTY\_CELL = ' ';**

**public static void initialize(char[][] board) {**

**for (int i = 0; i < 3; i++) {**

**for (int j = 0; j < 3; j++) {**

**board[i][j] = EMPTY\_CELL;**

**}**

**}**

**}**

**public static void printBoard(char[][] board) {**

**for (int i = 0; i < 3; i++) {**

**for (int j = 0; j < 3; j++) {**

**System.out.print(board[i][j] + "|");**

**}**

**System.out.println();**

**}**

**}**

**public static int checkFirstDiagonal(char[][] board, char ch) {**

**if (board[0][0] == ch && board[1][1] == ch && board[2][2] == ch) {**

**return 1;**

**} else {**

**return 0;**

**}**

**}**

**public static int checkSecondDiagonal(char[][] board, char ch) {**

**if (board[0][2] == ch && board[1][1] == ch && board[2][0] == ch) {**

**return 1;**

**} else {**

**return 0;**

**}**

**}**

**public static int checkFirstRow(char[][] board, char ch) {**

**if (board[0][0] == ch && board[0][1] == ch && board[0][2] == ch) {**

**return 1;**

**} else {**

**return 0;**

**}**

**}**

**public static int checkSecondRow(char[][] board, char ch) {**

**if (board[1][0] == ch && board[1][1] == ch && board[1][2] == ch) {**

**return 1;**

**} else {**

**return 0;**

**}**

**}**

**public static int checkThirdRow(char[][] board, char ch) {**

**if (board[2][0] == ch && board[2][1] == ch && board[2][2] == ch) {**

**return 1;**

**} else {**

**return 0;**

**}**

**}**

**public static int checkFirstColumn(char[][] board, char ch) {**

**if (board[0][0] == ch && board[1][0] == ch && board[2][0] == ch) {**

**return 1;**

**} else {**

**return 0;**

**}**

**}**

**public static int checkSecondColumn(char[][] board, char ch) {**

**if (board[0][1] == ch && board[1][1] == ch && board[2][1] == ch) {**

**return 1;**

**} else {**

**return 0;**

**}**

**}**

**public static int checkThirdColumn(char[][] board, char ch) {**

**if (board[0][2] == ch && board[1][2] == ch && board[2][2] == ch) {**

**return 1;**

**} else {**

**return 0;**

**}**

**}**

**public static int main(String[] args) {**

**char[][] board = new char[3][3];**

**int a, b;**

**char player\_turn = '1';**

**Scanner scanner = new Scanner(System.in);**

**initialize(board);**

**printBoard(board);**

**for (int i = 0; i < 9; i++) {**

**System.out.println("Player " + player\_turn + " Enter Row and Column range(0-2) at which you want to give Input( ): ");**

**a = scanner.nextInt();**

**b = scanner.nextInt();**

**board[a][b] = player\_turn;**

**printBoard(board);**

**if ( checkFirstDiagonal(board,player\_turn)==1 || checkSecondDiagonal(board,player\_turn)==1 ||**

**checkFirstRow(board,player\_turn)==1 || checkSecondRow(board,player\_turn)==1 ||**

**checkThirdRow(board,player\_turn)==1 || checkFirstColumn(board,player\_turn)==1 ||**

**checkSecondColumn(board,player\_turn)==1 || checkThirdColumn(board,player\_turn)==1 ) {**

**System.out.println("Player " + player\_turn + " Won the Game");**

**return 0;**

**}**

**else if( player\_turn=='1' ) player\_turn='2';**

**else player\_turn='1';**

**}**

**System.out.println("Match Draw");**

**return 0;**

**}**

**}**