***Date – 06/03/2020***

***//Abstract class 1***

import java.io.\*;

abstract class Sample //object cannot be created and abstract key word is necessary

{

void show()

{

System.out.println("Parent Class");

}

}

class AbstractClass1 extends Sample

{

public static void main(String[] arg)

{

//Sample obj1 = new Sample();

AbstractClass1 obj = new AbstractClass1();

obj.show();

}

}

***//Abstract Class Notes***

/\*

1- Abstract Method require abstract key word

2- We cannot define body of method

3- It should be present in abstract class

4- When we have to use overiding concept we go with abstract method

5- Abstract method always define in interfaces

\*/

***//Abstract class OverRiding***

import java.io.\*;

abstract class Sample //object cannot be created and abstract key word is necessary

{

abstract void show();

//{

// System.out.println("Parent Class");

//}

}

class AbstractClass1 extends Sample

{

void show()

{

System.out.println("Child Class");

}

public static void main(String[] arg)

{

AbstractClass1 obj = new AbstractClass1();

obj.show();

}

}

***//Abstract Class Overiding 2 Addition Substraction***

import java.io.\*;

class add

{

void func(int a, int b)

{

System.out.println("Addition is "+(a+b));

}

}

class sub extends add

{

void func(int a, int b)

{

super.func(a,b);

System.out.println("Substraction is "+(a-b));

}

}

class mar63

{

public static void main (String args[])throws IOException

{

sub ob=new sub();

ob.func(10,20);

}

}

***//Abstract Class Overiding 3***

***//Area and perimeter of Circle***

import java.io.\*;

abstract class Area

{

void CArea(double r)

{

System.out.println("Area: "+(3.14\*r\*r));

}

}

class Perimeter extends Area

{

void CPerimeter(double r)

{

super.CArea(r);

System.out.println("Perimeter: "+(6.23\*r));

}

}

class AbstractClass3

{

public static void main (String args[])throws IOException

{

double r;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter Radius: ");

r = Double.parseDouble(br.readLine());

Perimeter p = new Perimeter();

p.CPerimeter(r);

}

}

***//Abstract Class 4***

***//wap to define spy method as abstract method and find out all spy number between 1 to 1000 along with prime no***

import java.io.\*;

abstract class Niven

{

abstract void Niven1();

}

class Prime extends Niven

{

void Prime1()

{

int c, i, j;

for(i = 1; i <= 1000; i++)

{

c=0;

for(j = 2; j<=i/2; j++)

{

if(i%j==0)

{

c++;

}

}

if(c==0)

{

System.out.print(i+"\t");

}

}

System.out.println("");

}

void Niven1()

{

int sum=0, rem, num, i, j;

for(i = 1; i <= 1000; i++)

{sum=0;

num = i;

while(num>0)

{

rem = num % 10;

sum += rem;

num /= 10;

}

if(i%sum==0)

{

System.out.print(+i+"\t");

}

}

}

}

class AbstractClass4

{

public static void main (String args[])

{

Prime p = new Prime();

System.out.println("Prime Numbers: ");

p.Prime1();

System.out.println("Niven Numbers: ");

p.Niven1();

}

}

**//Interface 1**

**//InterfaceDemo**

interface a

{

public void add(int a, int b);

}

interface b

{

public void sub(int a, int b);

}

class InterfaceDemo implements a,b

{

public void add(int a, int b)

{

System.out.println("Addition: " + (a+b));

}

public void sub(int a, int b)

{

System.out.println("Subtraction: " + (a-b));

}

public static void main(String arg[])

{

InterfaceDemo ob = new InterfaceDemo();

ob.add(20,30);

ob.sub(50,10);

}

}

***//Interface 1***

***//wap to define volume of sphere and area with multiple inheritence***

import java.io.\*;

interface Volume

{

public void volume1(double r);

}

interface Area

{

public void area1(double r);

}

class Interface1 implements Volume, Area

{

public void volume1(double r)

{

System.out.println("Volume: " + ((1.33\*3.14)\*r\*r\*r));

}

public void area1(double r)

{

System.out.println("Area: " + ((4\*3.14)\*r\*r));

}

public static void main(String[] arg)throws IOException

{

double r;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter Radius: ");

r = Double.parseDouble(br.readLine());

Interface1 ob = new Interface1();

ob.volume1(r);

ob.area1(r);

}

}

***//Interface 2***

***//wap to show entered no is happy no or not, spy or not using multiple inheritence***

import java.io.\*;

interface Spy

{

public void spyf(int a);

}

interface Happy

{

public void happyf(int a);

}

class Interface2 implements Spy, Happy

{

public void spyf(int a)

{

int b=0,c=1;

for(int r; a>0; a=a/10)

{

r=a%10;

b=b+r;

c=c\*r;

}

if(b==c)

{

System.out.println("Spy Number");

}

else

{

System.out.println("NOT Spy Number");

}

}

public void happyf(int a)

{

int b,c=a;

for(;(c>1) && (c!=4); a=c)

{

b=0;

for(int r;a>0;a=a/10)

{

r=a%10;

b=b+(r\*r);

}

c=b;

if(c==1)

System.out.println("Happy Number");

else if(c==4)

System.out.println("Not a Happy Number");

}

}

public static void main(String[] arg)throws IOException

{

int a;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter Number: ");

a = Integer.parseInt(br.readLine());

Interface2 ob = new Interface2();

ob.spyf(a);

ob.happyf(a);

}

}

***Date – 08/03/2020***

***//Packages***

package PG2020;

public class ArithDemo

{

public double add(double a, double b)

{

System.out.println("Addition: " + (a + b));

return 0;

}

public double sub(double a, double b)

{

System.out.println("Subtraction: " + (a - b));

return 0;

}

public double multiply(double a, double b)

{

System.out.println("Addition: " + (a \* b));

return 0;

}

public int divide(double a, double b)

{

System.out.println("Addition: " + (a / b));

return 0;

}

}

import java.io.\*;

import PG2020.AreaVolume;

class PackageDemo

{

public static void main(String arg[]) throws IOException

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int a, b;

ArithDemo ar = new ArithDemo();

System.out.print("Enter value for Addition: ");

a = Integer.parseInt(br.readLine());

b = Integer.parseInt(br.readLine());

ar.add(a, b);

System.out.print("Enter value for Subtraction: ");

a = Integer.parseInt(br.readLine());

b = Integer.parseInt(br.readLine());

ar.sub(a, b);

System.out.print("Enter value for Multiplication: ");

a = Integer.parseInt(br.readLine());

b = Integer.parseInt(br.readLine());

ar.multiply(a, b);

System.out.print("Enter value for Division: ");

a = Integer.parseInt(br.readLine());

b = Integer.parseInt(br.readLine());

ar.divide(a, b);

}

}

***//Package 2***

package PG2020;

public class AreaVolume

{

public double areaCircle(double r)

{

System.out.println("Area of Circle: " + (3.14 \* r \* r));

System.out.println("");

return 0;

}

public double areaTriangle(double b, double h)

{

System.out.println("Area of Triangle: " + (0.5 \* b \* h));

System.out.println("");

return 0;

}

public double areaRectangle(double l, double b)

{

System.out.println("Addition: " + (l \* b));

System.out.println("");

return 0;

}

public double areaSphere(double r)

{

System.out.println("Addition: " + (4 \* 3.14 \* r \* r));

System.out.println("");

return 0;

}

public double volumeCircle(double r)

{

System.out.println("Perimeter of Circle: " + (2 \* 3.14 \* r));

System.out.println("");

return 0;

}

public double volumeSphere(double r)

{

System.out.println("Addition: " + (1.33 \* 3.14 \* r \* r \* r));

System.out.println("");

return 0;

}

}

import java.io.\*;

import PG2020.AreaVolume;

class PackageDemo2

{

public static void main(String arg[]) throws IOException

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

double r, l, b, h;

AreaVolume ar = new AreaVolume();

System.out.print("Enter radius for Area of Circle: ");

r = Double.parseDouble(br.readLine());

ar.areaCircle(r);

System.out.print("Enter Base and Height for Area of Triangle: ");

b = Double.parseDouble(br.readLine());

h = Double.parseDouble(br.readLine());

ar.areaTriangle(b, h);

System.out.print("Enter Length and Breadth for Area of Rectangle: ");

l = Double.parseDouble(br.readLine());

b = Double.parseDouble(br.readLine());

ar.areaRectangle(l, b);

System.out.print("Enter radius for Area of Sphere: ");

r = Double.parseDouble(br.readLine());

ar.areaSphere(r);

System.out.print("Enter Radius for Perimeter of Circle: ");

r = Double.parseDouble(br.readLine());

ar.volumeCircle(r);

System.out.print("Enter Radius for Volume of Sphere: ");

r = Double.parseDouble(br.readLine());

ar.volumeSphere(r);

}

}

***//Logical Numbers***

//Logical Number

package PG2020;

public class LogicalNumber

{

public void Prime(int a)

{

int c=0;

for(int i=2;i<=a/2;i++)

{

if(a%i==0)

{

c++;

}

}

if(c==0)

{

System.out.println("Prime Number");

}

else

{

System.out.println("Not a Prime Number");

}

}

public void Armstrong(int a)

{

int c=a,r,b;

for(b=0;c>0;c=c/10)

{

r=c%10;

b=b+(r\*r\*r);

}

if(a==b)

{

System.out.println("Armstrong Number");

}

else

{

System.out.println("Not a Armstrong Number");

}

}

public void Palindrome(int a)

{

int c=a;

int b=0;

for(int r;a>0;a=a/10)

{

r=a%10;

b=(b\*10)+r;

}

if(b==c)

{

System.out.println("Palindrome Number");

}

else

{

System.out.println("Not a Palindrome Number");

}

}

public void Spy(int a)

{

int c=0,b=1;

for(int r;a>0;a=a/10)

{

r=a%10;

b=b\*r;

c=c+r;

}

if(b==c)

{

System.out.println("Spy Number");

}

else

{

System.out.println("Not Spy number");

}

}

public void Niven(int a)

{

int b, c = 0;

a = c;

b = 0;

for (int r; a > 0; a = a / 10)

{

r = a % 10;

b = b + r;

}

if (c % b == 0)

{

System.out.println("Niven Number");

}

else

{

System.out.println("\nNot Niven Number");

}

}

public void Happy(int a)

{

int b,c=a;

for(;(c>1)&&(c!=4);a=c)

{

b=0;

for(int r;a>0;a=a/10)

{

r=a%10;

b=b+(r\*r);

}

c=b;

if(c==1)

System.out.println("Happy Number");

else if(c==4)

System.out.println("Not a Happy Number");

}

}

public void Magical(int a)

{

int c=a,b,d,r;

for(b=0,d=1;a>0;a=a/10)

{

r=a%10;

b=b+r;

d=d\*r;

}

if((b+d)==c)

{

System.out.println("Magical Number");

}

else

{

System.out.println("Not a Magical Number");

}

}

public void Neon(int a)

{

int c=a\*a,b,r;

for(b=0;c>0;c=c/10)

{

r=c%10;

b=b+r;

}

if(b==a)

{

System.out.println("Neon Number");

}

else

{

System.out.println("Not a Neon Number");

}

}

public void Perfect(int a)

{

int b=a,c=0;

for(int i=1;i<=a/2;i++)

{

if(a%i==0)

{

c=c+i;

}

}

if(c==b)

{

System.out.println("Perfect Number");

}

else

{

System.out.println("Not a Perfect Number");

}

}

public void Buzz(int a)

{

if((a%10==7)||(a%7==0))

{

System.out.println("Buzz Number");

}

else

{

System.out.println("Not a Buzz Number");

}

}

}

//WAP to create logical number package to display prime, armstrong, palindrome, spy, niven, happy, magical

import java.io.\*;

import PG2020.LogicalNumber;

class LogicalNumberMain

{

public static void main (String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader (System.in));

logic obj=new logic();

int a;

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

a=Integer.parseInt(br.readLine());

obj.Prime(a);

obj.Armstrong(a);

obj.Palindrome(a);

obj.Spy(a);

obj.Niven(a);

obj.Happy(a);

obj.Magical(a);

obj.Neon(a);

obj.Perfect(a);

obj.Buzz(a);

}

}

***Date – 14/03/2020***

***//wap to display entered no is perfect no or not, niven and buzz with help of menu drive and user friendly***

***//14 March 1***

import java.io.\*;

import PG2020.LogicalNumbers;

class PackageImport

{

public static void main(String[] arg) throws IOException

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

String ch;

do

{

int option, number;

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

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

System.out.println("1 for Perfect Number");

System.out.println("2 for Niven Number");

System.out.println("3 for Buzz Number");

System.out.print("Your Choice: ");

option = Integer.parseInt(br.readLine());

LogicalNumbers ln = new LogicalNumbers();

switch(option)

{

case 1:

System.out.println("Enter Number for Perfect Number: ");

number = Integer.parseInt(br.readLine());

ln.Perfect(number);

break;

case 2:

System.out.println("Enter Number for Niven Number: ");

number = Integer.parseInt(br.readLine());

ln.Niven(number);

break;

case 3:

System.out.println("Enter Number for Buzz Number: ");

number = Integer.parseInt(br.readLine());

ln.Buzz(number);

break;

default:

System.out.println("Invalid Input");

}

System.out.println("Want to Continue? Y/N: ");

ch = br.readLine();

}while(ch.equals("Y") || ch.equals("y"));

}

}

***5 Exception Handling Key Words***

***Try***

***Catch***

***Finally***

***Throws***

***Throw***

***//Exception Handling Demo TRY CATCH***

***//14 March 2***

import java.io.\*;

class ExceptionHandlingDemo

{

public static void main(String arg[])

{

try

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int a, b;

System.out.println("Enter 2 numbers: ");

a = Integer.parseInt(br.readLine());

b = Integer.parseInt(br.readLine());

System.out.println("A/b: " + (a/b));

}

catch(IOException e1)

{

System.out.println("e1");

}

catch(ArithmeticException e)

{

System.out.println("Divide by 0 ERROR: " + e);

}

}

}

***//Exception Handling 1***

***//wap to display arithmatical exception error and array index out of Bound exception with the help of multicatch block***

***//14 March 2***

import java.io.\*;

class ExceptionHandling1

{

public static void main(String arg[])throws IOException

{

try

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int a, b, c;

System.out.println("Enter 2 numbers: ");

a = Integer.parseInt(br.readLine());

b = Integer.parseInt(br.readLine());

System.out.println("A/b: " + (a/b));

c = Integer.parseInt(arg[0]);

System.out.println("Multiply: " + (a\*c));

}

catch(ArrayIndexOutOfBoundsException e1)

{

System.out.println(e1);

}

catch(ArithmeticException e)

{

System.out.println(e);

}

}

}

***//MultiThreading***

***Thread <- Inheritence <- Classes***

***Runnable <- Interface <- Interface***

***//Life Cycle of Thread***

***1- Start -> start(),run()***

***2- Interupt-> sleep(),suspend()***

***3- Destroy***

***//Multi Threading Demo***

import java.io.\*;

class thr extends Thread

{

public void runf()throws Exception

{

for(int i=0;i<20;i++)

{

System.out.println("i= "+i);

Thread.sleep(500);

}

}

}

class MultiThreadingDemo extends Thread

{

public void runf()throws Exception

{

for(int j=0;j<20;j++)

{

System.out.println("j= "+j);

}

}

public static void main (String args[])throws Exception

{

MultiThreadingDemo obj=new MultiThreadingDemo();

thr ob1=new thr();

ob1.runf();

obj.runf();

obj.start();

ob1.start();

}

}

***//Multi Threading 1***

***//Sleep()***

import java.io.\*;

class A extends Thread

{

public void show()throws InterruptedException

{

for(int i=0; i<5; i++)

System.out.println("i: " + i);

Thread.sleep(1500);

}

}

class B extends Thread

{

public void output()throws InterruptedException

{

for(int j=0; j<20; j++)

System.out.println("j: " + j);

Thread.sleep(500);

}

}

class MultiThreading1

{

public static void main(String arg[])throws InterruptedException

{

A ob = new A();

B obj = new B();

ob.show();

obj.output();

}

}

***//Multi Threading 2***

***//Interface***

import java.io.\*;

class A extends Thread implements Runnable

{

public void run()

{

try

{

for(int i=0; i<5; i++)

System.out.println("i: " + i);

Thread.sleep(500);

}

catch(Exception e)

{

System.out.println(e);

}

}

}

class B extends Thread implements Runnable

{

public void run()

{

try

{

for(int j=0; j<10; j++)

System.out.println("j: " + j);

Thread.sleep(500);

}

catch(Exception e1)

{

System.out.println(e1);

}

}

}

class MultiThreading2

{

public static void main(String arg[])throws InterruptedException

{

A ob = new A();

B obj = new B();

ob.start();

obj.start();

}

}