1.Create class box and box3d bax3d is extended class of box. The above two classes going to pull fill following requirement Include constructor -set value

of length, breadth, height Find out area and volume. Note: Base class and sub classes have respective methods and instance variables.

class Box

{

private int length;

private int breadth;

public Box ( )

{

length =0;

breadth =0;

}

public Box (int x, int y)

{

length = x;

breadth =y;

}

public void setvalue (int x, int y)

{

length = x;

breadth = y;

public int area ( )

{

return (length \* breadth);

}

}

class Box3d extends Box

{

private int height;

public Box3d ( )

{

super ( );

height = 0;

}

public Box3d(int x, int y, int z)

{

super (x, y);

height = z;

}

public void setvalue (int x, int y, int z)

{

super.setvalue(x, y);

height = z;

}

public int volume ( )

{

return (super.area ( ) \* height );

}

public static void main(String arga [ ])

{

Box b1 = new Box ( );

Box3d b2 = new Box3d(14,36,16);

b1.setvalue (30,35);

System.out.println (“The area of b1 is : “ + b1.area( ) );

System.out.println (“The volume of b2 is : “+ b2.volume ( ));

}

}

Create class Number with only one private instance variable as a double primitive type.

To include the following methods (include respective constructors) isZero(), isPositive(), isNegative(), isOdd(), isEven().

isPrime(), isAmstrong) the above methods return boolean primitive type getFactorial().getSqrt(), getSqr(), sumDigits().getReverse()

the above 2 methods return double primitive type void listFactor), void dispBinary().

import java.lang.Math. \*;

import java.lang.Number.\*;

class Num

{

private double db1;

private long lg;

public Num ( )

{

db1 = 108.0d;

lg = 249;

}

public Num4(double d, long l)

{

db1 = d;

lg = 1;

}

public boolean isZero ( )

{

if (db1 == 0.0)

return true;

else

return false;

}

public boolean isPositive ( )

{

if(db1 > 0.0)

return true;

else

return false;

}

public boolean isNegative ( )

{

if (db1 < 0.0)

return true ;

else

return false;

}

public boolean isodd( )

{

if (db1 % 2 != 0.0)

return true;

else return false;

}

public boolean isEven ( )

{

if (db1 % 2 == 0.0)

return true ;

else return false;

}

public boolean isPrime ( )

{

int i, lastn;

double a;

boolean flag;

a = Math.sqrt(lg);

lastn = (int)a;

flag = true;

for (i=2; i<lastn; i++)

{

if (lg != i)

{

if( lg % i ==0)

{

flag = false;

break;

}

}

}

if (flag)

return true;

else return false;

}

public boolean isAmstrong ( )

{

if (db1 == 0.0)

return true;

else return false;

}

public double getFactorial ( )

{

double d=1;

for(int i = 1;i 9)

{

d += lg % 10;

lg = lg/10;

}

d +=lg;

return d;

}

public double getReverse ( )

{

double d =0;

double temp;

while (lg>9)

{

temp = lg%10;

d = d \* 10 + temp;

lg = lg/10;

System.out.println (“\n”+ temp + “\t” + d +” \t “+lg);

}

d = d \* 10 +lg;

System.out.println (“Inside class” + d);

return d;

}

public void dispBinary ( )

{

System.out.println(“ByteValue of lg :” + Long.toBinaryString(lg));

}

public static void main (String args [ ])

{

Num4 mynum = new Num4( );

double d = 199;

System.out.println(” The given numbers are 108.0d and 249″);

System.out.println (“isZero ” + mynum.isZero() );

System.out.println (“isPositive ” + mynum. isPositive());

System.out.println (“isNegative ” + mynum.isNegative() );

System.out.println (“isOdd ” + mynum.isodd());

System.out.println (“isEven ” + mynum.isEven());

System.out.println (” isPrime ” +mynum.isPrime());

System.out.println (“getFactorial ” + mynum.getFactorial());

System.out.println (“getSqrt ” + mynum. getSqrt( ));

System.out.println (“getSqr ” + mynum.getSqr( ) );

System.out.println (“sumDigits ” + mynum.sumDigits( ));

System.out.println (“getReverse ” + mynum.getReverse( ));

mynum.dispBinary();

System.out.println (” isPrime ” +mynum.isPrime());

}

}

Write a program to create a class named shape. In this class we have three sub classes circle, triangle and square

each class has two member function 3 named draw 0) and erase () Create these using polymorphism concepts.

class Shape

{

void draw()

{

System.out.println(“Shape draw()”);

}

void erase()

{

System.out.println (“ Shape erase()”);

}

}

class Circle extends Shape

{

void draw()

{

System.out.println (“Circle draw()”);

}

void erase()

{

System.out.println (“Circle erase()”);

}

}

class Triangle extends Shape

{

void draw()

{

System.out.println(“Triangle erase()”);

}

}

class Square extends Shape

{

void draw()

{

System.out.println (“Square draw()”);

}

void erase()

{

System.out.println (“Square erase()”);

}

}

public class Shapes

{

public static Shape randshape()

{

switch((int)(Math.random()\*3))

{

case 0: return new Circle();

case 1: return new Square();

case 2: return new Triangle();

default : System.out.println(“default”);

return new Shape();

}

}

public static void main (String arg[])

{

Shape s[] = new Shape[9];

for(int i = 0;i< s.length; i++) s[i] = randshape(); for(int i= 0;i < s.length; i++) s[i].draw(); } }

Find the Fibonacci numbers are defined as F \* 0 = 1, F \* 1 = 1 and Fi = Fi - 1 + Fi - 2

public class Fibonacci

{

long fib (int n)

{

if (n <= 1) return 1;

else

return (fib(n – 1) + fibo (n –2) );

}

public static void main (String arg [ ])

{

Fibonacci f;

long l;

f = new Fibonacci ( );

l = f.fib (4);

System.out.println (“4th Fibonacci number is: “+1);

}

}

Find the maximum of an array. Let a[] be an array of integers. if n= 1, a[0] is the only number in the array and so, maximum = a[0]. if n > 1, then do the following: find the maximum of n-1 entries of the array.

Compare this maximum with the last entry a[n-1] and finalize.

public class Factorial

{

int fact (int n)

{

if (n <= 1) return 1;

else return ( n \* fact(n – 1)) ;

}

public static void main (String arg [ ])

{

int fa, fb, fc;

int a = 6, b = 7, c = 8;

Factorial f;

f = new Factorial ( );

fa = f.fact (a);

fb =f.fact (b);

fc = f.fact (c);

System.out.println(“Factorial of “+ a + “is” + fa);

System.out.println (“Factorial of “+ b + “ is” + fb);

System.out.println(“Factorial of “+ c + “is” + fc);

}

}