

INDEX

| Sl. No. | Name of Program | Date | Page No. | Remarks & Signature |
|------------|--|----------|-------------|------------------------|
| 1 | Bigest among three numbers | 8/3/18 | 1-2 | |
| 2 | Armstrong or not | 8/3/18 | 3 | |
| 3 | palindrome | 13/3/18 | 4 | |
| 4 | factorial of a given number | 15/3/18 | 5 | |
| 5 | fibonacci series | 15/3/18 | 6 | |
| 6 | Giving i/p through commandline arguments | 21/3/18 | 7 | |
| 7 | concatination of 2 words provided during runtime | 28/3/18 | 8 | |
| 8 | find the area of rectangle. using constructor | 16/4/18 | 9 | |
| 9 | Addition of 2 matrices | 17/4/18 | 10-11 | |
| 10 | Implementing Multiple inheritance | 19/4/18 | 12-14 | |
| 11 | Example of interface | 24/4/18 | 15-16 | |
| 12 | Application of two dimensional array | 29/4/18 | 17 | |
| 13 | Sorting a list of numbers using array | 10/5/18 | 18-19 | |
| 14 | Defining a subclass | 10/5/18 | 20-21 | |
| 15 | package | 11/5/18 | 22 | |
| 16 | Multi-threading | 23/5/18 | 23-24 | |
| 17 | A pgm .using runnable interface | 11/06/18 | 25 | |
| 18 | Applet programming | 11/6/18 | 26 | |
| 19 | Applet programs to draw a human face | 21/6/18 | 27-28 | |
| 20 | Drawing Lines and Rectangles | 29/6/18 | 29-30 | |
| 21 | Drawing polygons | 12/7/18 | 31-32 | |
| 22 | | | | |
| 23 | | | | |
| 24 | | | | |
| 25 | | | | |
| 26 | | | | |
| 27 | | | | |
| 28 | | | | |
| 29 | | | | |
| 30 | | | | |
| 31 | | | | |
| 32 | | | | |

Completed

Met

Output

D:\> uit5.java Biggest.java

D:\> uit5.java Biggest

Enter the 1st number:

10

Enter the 2nd number:

20

Enter the 3rd number:

30

30 is greater

PROGRAM NO:1

BIGGEST AMONG THREE NUMBERS

write a java program to find the biggest among three numbers.

```
Program: import java.util.Scanner;  
class Biggest  
{  
    public static void main(String args[])  
    {  
        Scanner obj = new Scanner(System.in);  
        int a, b, c;  
        System.out.println("Enter the 1st number:");  
        a = obj.nextInt();  
        System.out.println("Enter the 2nd number:");  
        b = obj.nextInt();  
        System.out.println("Enter the 3rd number:");  
        c = obj.nextInt();  
        if (a > b & a > c)  
        {  
            System.out.println(a + " is greater");  
        }  
        else if (b > c)  
        {  
            System.out.println(b + " is greater");  
        }  
        else  
        {  
        }
```

③

③

③

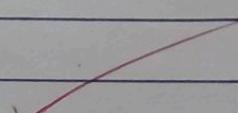
8

System.out.println("C '+' is greater');

}

{

{



OUTPUT

D:\javas>javac Armstrong.java

D:\VIT>java Armstrong

Enter a number 153
Armstrong

Entered a number 154
not an Armstrong.

A

(5)

3

PROGRAM NO: 2

AMSTRONG OR NOT

write a program to find out the given number is an amstrong or not

```
Program : import java.util.Scanner;
```

```
class Armstrong
```

{

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

{

```
Scanner obj = new Scanner (System.in);
```

```
int n, s=0, r=0, t;
```

```
System.out.println ("Enter a number:");
```

```
n=obj.nextInt();
```

```
t=n;
```

```
while (n>0)
```

{

```
r=n%10;
```

```
n=n/10;
```

```
s=s+(r*r*r);
```

{

```
if (t==s)
```

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

```
else
```

{

```
System.out.println ("Not an amstrong");
```

{

{

{

OUTPUT

D:\VITS>javac palindrome.java

D:\VITS>java palindrome

Enter a number: 321

Not a palindrome

PROGRAM NO: 3

PALINDROME OR NOT

Write a program to check whether the given number is palindrome or not

program :

```
import java.util.Scanner;  
class palindrome  
{  
    public static void main (String args[])  
    {  
        Scanner obj = new Scanner (System.in);  
        int n, s=0, r=0, t;  
        System.out.println ("Enter a number :");  
        n = obj.nextInt();  
        t = n;  
        while (n>0)  
        {  
            r = n % 10;  
            n = n / 10;  
            s = (s * 10) + r;  
        }  
        if (t == s)  
            System.out.println ("palindrome");  
        else  
            System.out.println ("not a palindrome");  
    }  
}
```

OUTPUT

D:\UITS>javac factorial.java

D:\UITS>java factorial

Enter a number: 6

The factorial is: 720

(a)

(1)

(1)

(2)

PROGRAM NO-4

FACTORIAL OF A NUMBER

Write a program to find the factorial of a given number

Program :

import java.util.Scanner;

class factorial

{

 public static void main (String args[])

{

 Scanner obj = new Scanner (System.in);

 int n, f=1;

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

 n = obj.nextInt();

 System.out.println ("The factorial is :");

 while (n > 0)

{

 f=f*n;

 n--;

}

 System.out.println (f);

}

}

OUTPUT

D:\javavits>javae fibinocijava

D:\javavits>java fibinocj

Enter a number : 5

The fibinocj Series are:

0

1

2

3

8

PROGRAM NO: 5

FIBONACI SERIES

Write a program to display Fibonacci Series upto a particular range.

Program :

```
import java.util.Scanner;
class fibonacci
{
    public static void main (String args[])
    {
        Scanner obj = new Scanner (System.in);
        int n, a=0, b=1, c;
        System.out.println ("Enter the number:");
        n=obj.nextInt();
        System.out.println (a);
        System.out.println (b);
        while (n>2)
        {
            c=a+b;
            a=b;
            b=c;
            n--;
        }
        System.out.println (c);
    }
}
```

12

OUTPUT

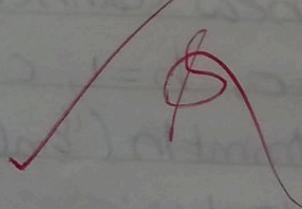
D:\UUT>javac comment.java

D:\UUT>java comment 24 76

Enter the 1st number : 24

Enter the 2nd number : 76

$$\text{Sum} = 100$$



(13)

PROGRAM No:6

13

7

GIVE INPUT THROUHT COMMENT LINE ARGUMENTS

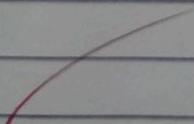
Wrie a program of giving input through comment line arguments.

Program :

```
class comment
{
    public static void main (String args[])
    {
        int a, b, c;
        a = Integer.parseInt (args[0]);
        System.out.println ("first number = "+a);
        b = Integer.parseInt (args[1]);
        System.out.println ("second number = "+b);
        c = a+b;
        System.out.println ("sum = "+c);
    }
}
```

}

}



OUTPUT

D:\UIT> javac String.java

D:\UIT> java String "Saji" "tha"

Enter the 1st word: Saji

Enter the 2nd word: tha

Concatenated string: Sajitha

(15)

PROGRAM NO: 7

CONCATINATION OF TWO WORDS PROVIDED DURING
RUNTIME

write a program to concatenate two words that is provided during runtime.

Program:

class Concat

{

public static void main (String args[])

{

String a,b,c;

a = args[0];

System.out.println ("First word : " + a);

b = args[1];

System.out.println ("Second word : " + b);

c = a + b;

System.out.println ("Concatinated String : " + c);

}

2

(16)

Task Area

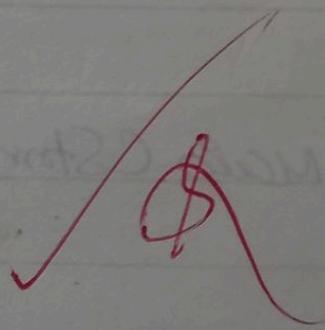
(16)

OUTPUT

D:\VIT>javac rectanglearea.java

D:\VIT>java rectanglearea

area = 150



(17)

PROGRAM NO:8

AREA OF RECTANGLE USING CONSTRUCTOR

Write a program to find area of rectangle using constructor

Program :

class Rectangle

{

 int length, width;

 Rectangle (int x, int y)

{

 length = x;

 width = y;

}

 int rectarea()

{

 return (length * width);

}

}

class Rectanglearea

{

 public static void main (String args[])

{

 Rectangle rect1 = new Rectangle (15, 10);

 int area1 = rect1.rectarea();

 System.out.println ("Area = "+ area1);

}

}

(A)

OUTPUT

D:\VIT> javac Addition.java

D:\VIT> java Addition

Enter the number of rows and columns of matrix: 2 2

Enter the element of 1st matrix: 1 2
3 4

Enter the element of 2nd matrix: 5 6
7 8

Sum of Entered Matrix: 6 8

10 12

(9)

PROGRAM NO : 9

ADDITION OF TWO MATRICES

write a program to add two matrices

program : import java.util.Scanner;
class addition

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

```
{  
    int x, c, i, j;
```

```
Scanner obj = new Scanner (System.in);
```

```
System.out.println ("Enter the no. of rows  
and columns of matrix:");
```

```
x = obj.nextInt();
```

```
c = obj.nextInt();
```

```
int a[][] = new int [x][c];
```

```
int b[][] = new int [x][c];
```

```
int sum[][] = new int [x][c];
```

```
System.out.println ("Enter the elements of 1st  
matrix:");
```

```
for (i=0; i<x; i++)
```

```
{
```

```
    for (j=0; j<c; j++)
```

```
{
```

```
        a[i][j] = obj.nextInt();
```

```
}
```

```
}
```

```
System.out.println ("Enter the elements of 2nd  
matrix:");
```

(20)

```
for(i=0; i<x; i++)
```

{

```
    for(j=0; j<c; j++)
```

{

```
        bc[i][j] = obj.nextInt();
```

{

{

```
System.out.println("Sum of entered matrices:");
```

```
for(i=0; i<x; i++)
```

{

```
{    for(j=0; j<c; j++)
```

{

```
        sum[i][j] = a[i][j] + b[i][j];
```

```
System.out.println(sum[i][j] + "lt");
```

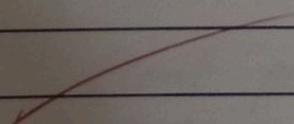
{

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

{

{

{



OUTPUT

D:\ UTS>javac hybrid.java

D:\ UTS>java hybrid

Roll no : 36

Marks obtained

Part 1 = 27.5

Part 2 = 33.0

Part wt = 6.0

Total score = 66.5

12

22

PROGRAM NO: 10

IMPLEMENTING MULTIPLE INHERITANCE

write a program of implementing multiple inheritance

Program :

```
class Student
```

```
{
```

```
int rollnumber;
```

```
void getnumber (int n)
```

```
{
```

```
rollnumber = n;
```

```
}
```

```
void putnumber()
```

```
{
```

```
System.out.println ("Roll no = " + rollnumber);
```

```
{
```

```
{
```

```
class Test extends Student
```

```
{
```

```
float pmt1, pmt2;
```

```
void getmarks (float m1, float m2)
```

```
{
```

```
pmt1 = m1;
```

```
pmt2 = m2;
```

```
{
```

```
void putmarks()
```

```
{
```

(23)

```
System.out.println ("Marks obtained");
```

```
System.out.println ("part1 = " + part1);
```

```
System.out.println ("part2 = " + part2);
```

{

{

interface Spots

{

```
float spotwt = 6.0f;
```

```
void putwt();
```

{

class Results extends Test implements Spots

{

```
float total;
```

```
public void putwt()
```

{

```
System.out.println ("spot wt = " + spotwt);
```

{

void display()

{

```
total = part1 + part2 + 8 * spotwt;
```

```
putnumber();
```

```
putmarks();
```

```
putwt();
```

```
System.out.println ("Total Score = " + total);
```

{

{

~~class Hybrid~~

{

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

{

(24)

Results student1 = new Results();
student1.setNumber(1234);
student1.setMarks(27.5f, 33.0f);
student1.display();

}

}

2f

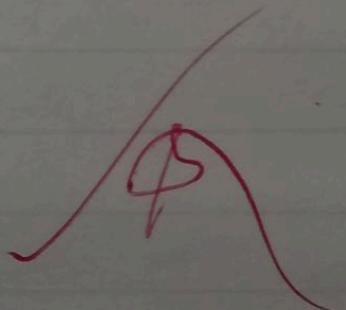
OUTPUT

D:\UIT>javae interfaceTest.java

D:\UIT>java interfaceTest

Area of Rectangle = 200.0

Area of Circle = 314.0



(26)

PROGRAM NO: 11

26

Example of Interface

write a program that defines an interface

program :

interface Area

{

final static float pi = 3.14f;

float compute (float x, float y);

{

class Rectangle implements Area

{

public float compute (float x, float y)

{

return (x*y);

{

{

class Circle implements Area

{

public float compute (float x, float y)

{

return (pi*x*x);

{

{

class Interface Test

{

public static void main (String args [])

{

27

16

Rectangle rect = new Rectangle();

Circle cir = new Circle();

Area area;

area = rect;

System.out.println ("Area of Rectangle = " +
area.compute (10, 20));

area = cir;

System.out.println ("Area of Circle = " +
area.compute (10, 0));

}

}

(28)

OUTPUT

MULTIPLICATION TABLE

| | | | | |
|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 |
| 2 | 4 | 6 | 8 | 10 |
| 3 | 6 | 9 | 12 | 15 |
| 4 | 8 | 12 | 16 | 20 |
| 5 | 10 | 15 | 20 | 25 |
| 6 | 12 | 18 | 24 | 30 |
| 7 | 14 | 21 | 28 | 35 |
| 8 | 16 | 24 | 32 | 40 |
| 9 | 18 | 27 | 36 | 45 |
| 10 | 20 | 30 | 40 | 50 |

| | | | | |
|----|----|----|----|-----|
| 6 | 7 | 8 | 9 | 10 |
| 12 | 14 | 16 | 18 | 20 |
| 18 | 21 | 24 | 27 | 30 |
| 24 | 28 | 32 | 36 | 40 |
| 30 | 35 | 40 | 45 | 50 |
| 36 | 42 | 48 | 54 | 60 |
| 42 | 49 | 56 | 63 | 70 |
| 48 | 56 | 64 | 72 | 80 |
| 54 | 63 | 72 | 81 | 90 |
| 60 | 70 | 80 | 90 | 100 |

(29)

PROGRAM NO: 12

14

APPLICATION OF TWO DIMENSIONAL ARRAY

write a program to show the application of two dimensional arrays.

program:

class MultiTable

{

final static int Rows = 11;

final static int Columns = 11;

public static void main (String args[])

{

int product [] [] = new int [Rows] [Columns];

int row, column;

System.out.println ("MULTIPLICATION TABLE");

System.out.println (" ");

int i, j;

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

{

for (j=1; j < Columns; j++)

{

product [i] [j] = i * j;

System.out.println (" " + product [i] [j]);

{

System.out.println (" ");

{

{

{

OUTPUT

D:\UIT>javac numbersorting.java

D:\UIT>java Numbersorting

Given list :

55

40

80

65

71

Sorted list :

80

71

65

55

40

✓ \$

(31)

16

(3)

PROGRAM NO:13

SORTING A LIST OF NUMBERS USING ARRAY

Write a program to sort a list of numbers using array.

Program :

```

class numberSorting
{
    public static void main (String args[])
    {
        int number [] = {55, 40, 30, 65, 71};
        int n = number.length;
        System.out.println ("Given List:");
        for (int i=0; i<n; i++)
        {
            System.out.println (" " + number[i]);
        }
        System.out.println ("(");
        for (int i=0; i<n; i++)
        {
            for (int j=i+1; j<n; j++)
            {
                if (number[i] < number[j])
                {
                    int temp = number[i];
                    number[i] = number[j];
                    number[j] = temp;
                }
            }
        }
    }
}

```

32

{

{

{

```
System.out.println ("Sorted list :");
```

```
for (int i=0; i<n; i++)
```

{

```
    System.out.println (" " + number[i]);
```

{

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

{

{

(33)

33

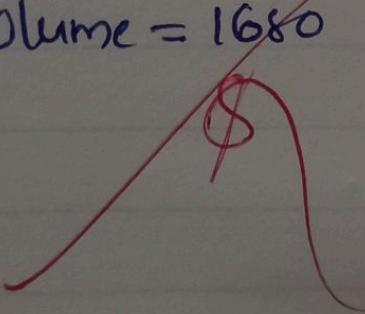
OUTPUT

D:\javait>javac isheatst.java

D:\javait>java isheatst

Area = 168

Volume = 1680



34

34

20

PROGRAM NO : 14

DEFINING A SUBCLASS

Write a program to define a subclass.

program:

class room

{

int length;

int breadth;

room (int x, int y)

{

length = x;

breadth = y;

}

int area()

{

return (length * breadth)

}

{

class bedroom extends room

{

int height;

bedroom (int x, int y, int z)

{

super (x, y);

height = z;

{

int volume()

{

(3)

return (length * breadth * height);

{

{

class Room

{

public static void main (String args[])

{

bedroom room1 = new bedroom (14, 12, 10);

int area1 = room1. area();

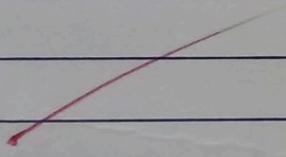
int volume1 = room1. volume();

System. out. println ("Area = "+ area1);

System. out. println ("Volume = "+ volume1);

{

{



36

OUTPUT

D:\UIT> cd pack

D:\UIT> pack>javac one.java

D:\UIT> pack>cd ..

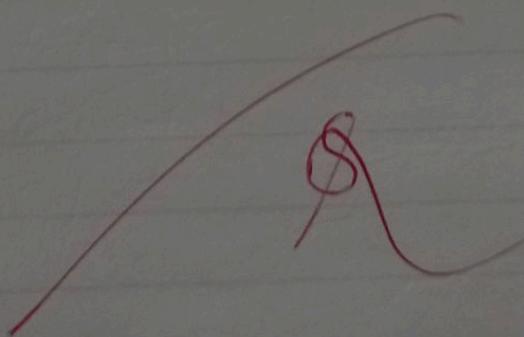
D:\UIT> javac Two.java

D:\UIT> java Two

S4BSC

UIT

PATHANAPURAM



(38)

PROGRAM alo:15

PACKAGE

Write a program that defines a package

program :

notebook1:

```
package abc;
public class one
{
```

```
    public void display()
{
```

```
        System.out.println("SABSC");
```

```
        System.out.println("UIT");
```

```
        System.out.println("Pathanapuram");
```

}

}

notebook2

import abc.one;

class two

{

public static void main (String args[])
{

one obj = new one();

obj.display();

}

}

(3)

OUTPUT

D:\UIT>javae ThreadTest.java

D:\UIT>java ThreadTest

from A:i=1

from A:i=2

from A:i=3

from B:j=1

from A:i=4

from C:k=1

from A:i=5

from B:j=2

exit A

from C:k=2

from B:j=3

from C:k=3

from B:j=4

from C:k=4

from B:j=5

exit B

from C:k=5

~~exit c.~~

(39)

PROGRAM NO: 16

MULTI THREADING

Write a program that defines multi-threading

Program :

class A extends Thread

{

public void run()

{

for (int i=1; i<=5; i++)

{

System.out.println("It from A: i = " + i);

{

System.out.println("Exit A");

{

{

class B extends Thread

{

public void run()

{

for (int j=1; j<=5; j++)

{

System.out.println("It from B: j = " + j);

{

System.out.println("Exit B");

{

{

(40)

class c extends thread

{

public void run()

{

for (int k=1; k<=5; k++)

{

System.out.println ("It from c: k = " + k);

{

System.out.println ("Exit c");

{

class ThreadTest

{

public static void main (String args[])

{

new A().start(); A a = new A();

new B().start(); B b = new B();

new C().start(); C c = new C();

{

b.start();

{

C c = new C();

c.start();



OUTPUT

(W)

D:\UIT>java RunnableTest.java

D:\UIT>java RunnableTest

end of Main thread

Thread x:1

Thread x:2

Thread x:3

Thread x:4

Thread x:5

Thread x:6

Thread x:7

Thread x:8

Thread x:9

Thread x:10

end of thread x

Ø

(L8)

PROGRAM NO:17

RUNNABLE INTERFACE

Write a program using runnable interface

Program :

class X implements Runnable

{

public void run()

{

for (int i = 1; i <= 10; i++)

{

System.out.println("It Thread X :" + i);

}

System.out.println("End of thread X");

{

{

class RunnableTest

{

public static void main (String args[])

{

Runnable r = new X();

Thread threadX = new Thread (Runnable r);

threadX.start();

System.out.println("End of main thread");

{

{

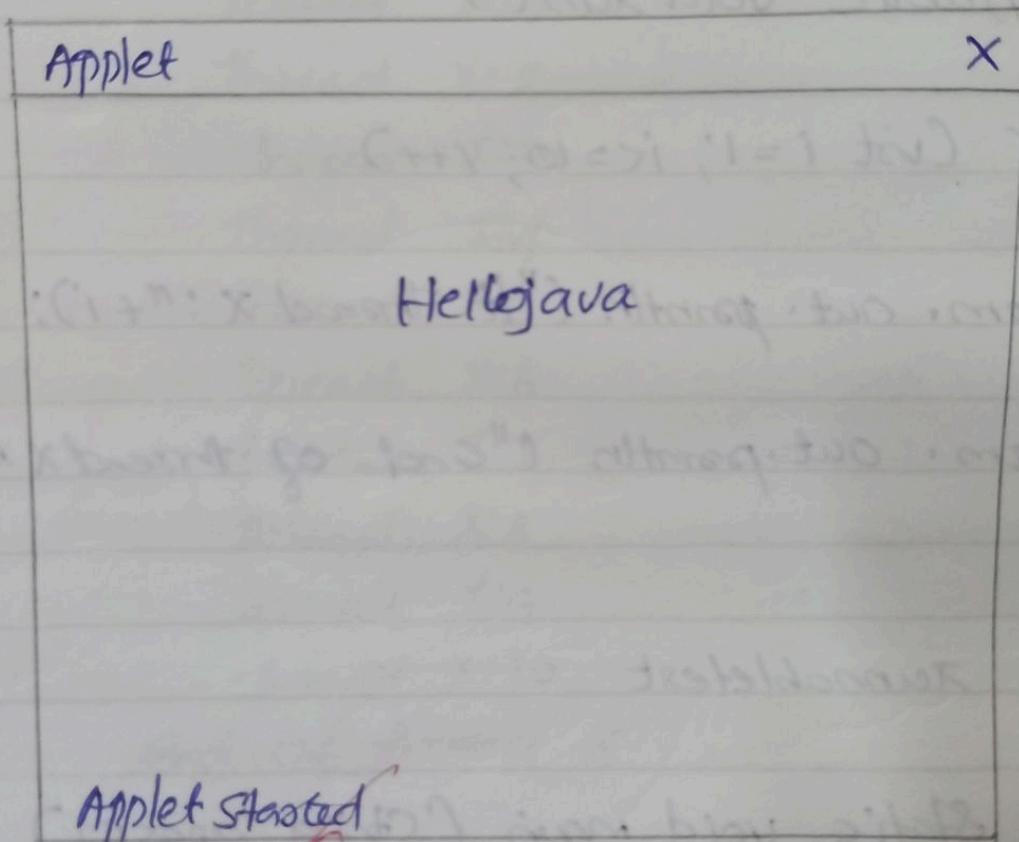
{



OUTPUT

D:\1 UIT>javac Hellojava.java

D:\1 UIT>appletviewer Hellojava.html



(W)

PROGRAM NO-18

APPLET PROGRAMMING

Write a java program for applet programming

Program :

Notepad - 1

```

import java.awt.*;
import java.applet.*;
public class Hellojava extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Hellojava", 10, 100);
    }
}

```

Notepad - 2

```

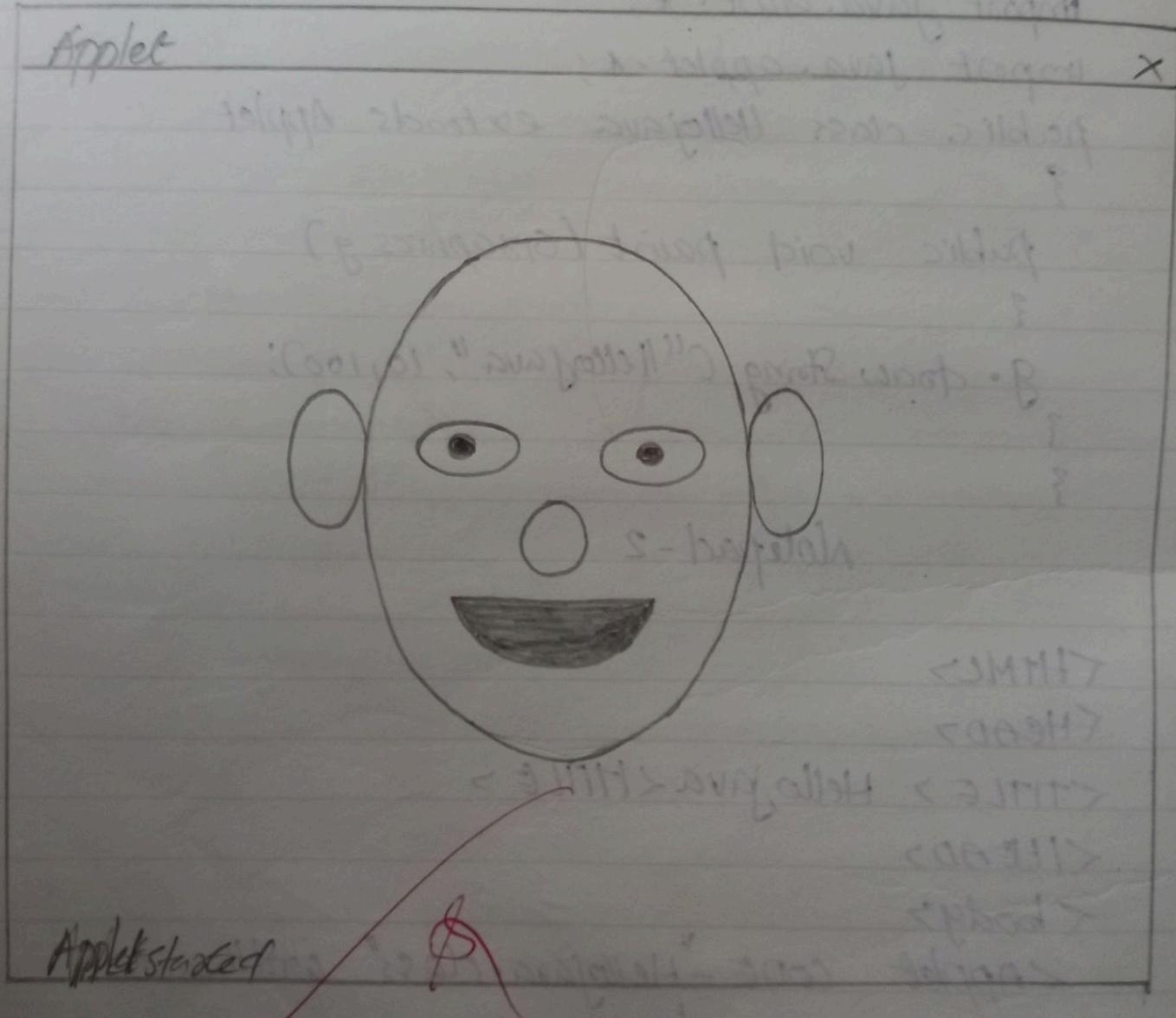
<HTML>
<HEAD>
<TITLE> Hellojava </TITLE>
</HEAD>
<body>
<Applet CODE = "Hellojava.class" width=100,
height=50>
</Applet>
</body>
</HTML>

```

OUTPUT

D:\UIT>javac face.java

D:\UIT>Appletviewer face.html



Applet started

(H6)

PROGRAM NO:19

APPLET PROGRAMMING
FACE

write an applet program to draw a human face.

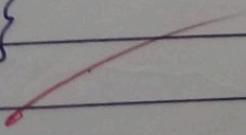
program:

notepad-1

```

import java.awt.*;
import java.applet.*;
public class face extends Applet
{
    public void paint(Graphics g)
    {
        g.drawOval(40, 40, 120, 150);
        g.drawOval(57, 75, 80, 20);
        g.drawOval(110, 75, 30, 20);
        g.fillOval(68, 81, 10, 10);
        g.FillOval(121, 81, 10, 10);
        g.drawOval(85, 100, 30, 30);
        g.fillRect(60, 125, 80, 40, 180, 180);
        g.drawOval(25, 92, 15, 30);
        g.drawOval(160, 92, 15, 30);
    }
}

```



28

17

notepad-2

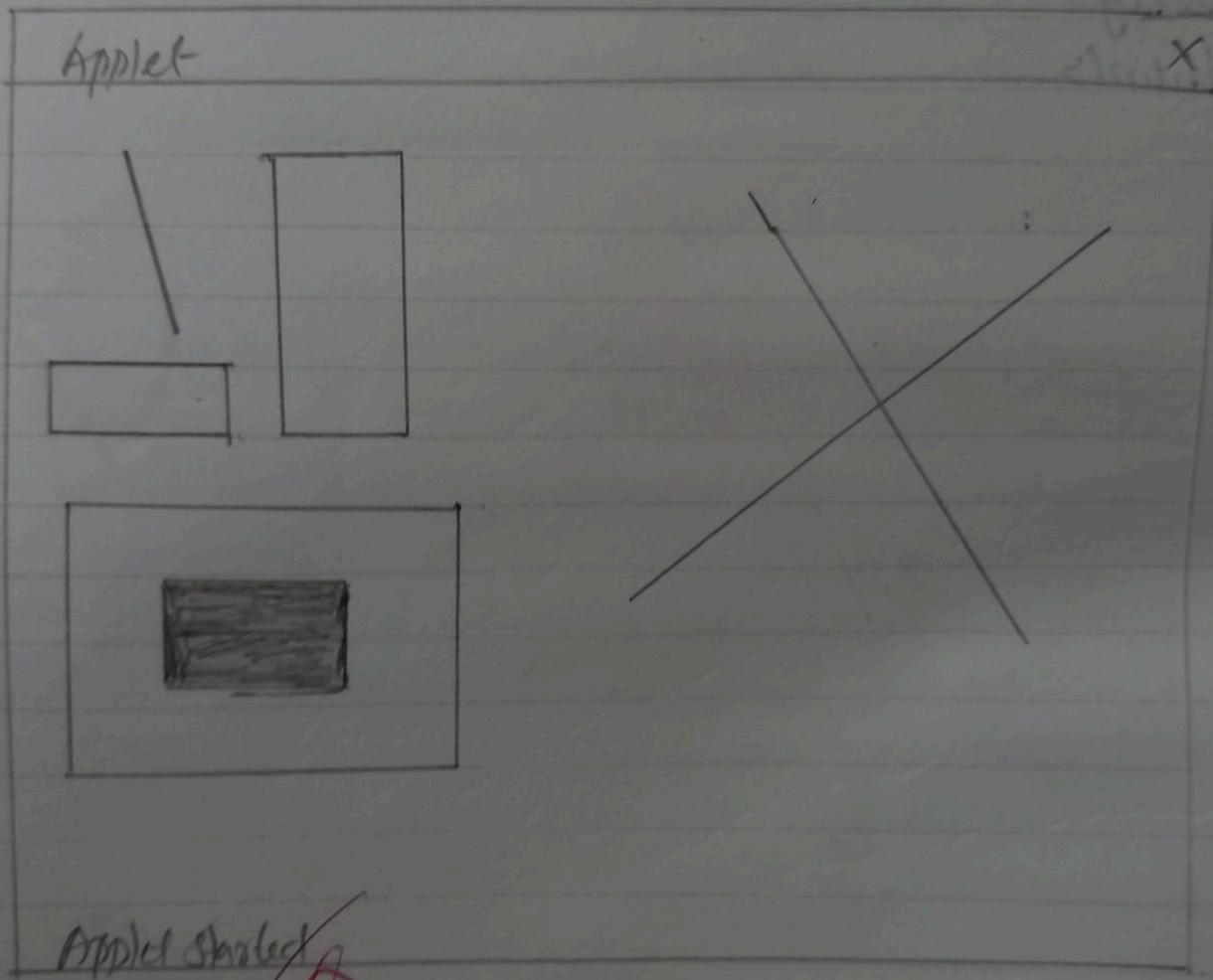
```
<HTML>
<head>
<title> face </title>
</head>
<body>
<Applet code="face.class" width=100 , height=50>
</Applet>
</body>
</HTML>
```

VB

OUTPUT

D:\UITS\javae LineRect.java

D:\UITS\appletviewer LineRect.html



(19)

PROGRAM NO: 20

DRAWING LINES AND RECTANGLES

Write an applet program to draw Lines and Rectangles.

Program :

Notepad-1

```

import java.awt.*;
import java.applet.*;

public class LineRect extends Applet
{
    public void paint(Graphics g)
    {
        g.drawLine(10, 10, 50, 50);
        g.drawRect(10, 60, 40, 30);
        g.fillRect(60, 10, 30, 80);
        g.drawRoundRect(10, 100, 60, 50, 10, 10);
        g.fillRoundRect(20, 110, 60, 30, 5, 5);
        g.drawLine(100, 10, 230, 140);
        g.drawLine(100, 140, 230, 10);
    }
}

```

Notepad-2

```

<html>
<head>
<title> LineRect </title>
</head>

```

30

(50) <body>

<Applet code = "LineRect.class" width=250,
height=200>

</Applet>

</body>

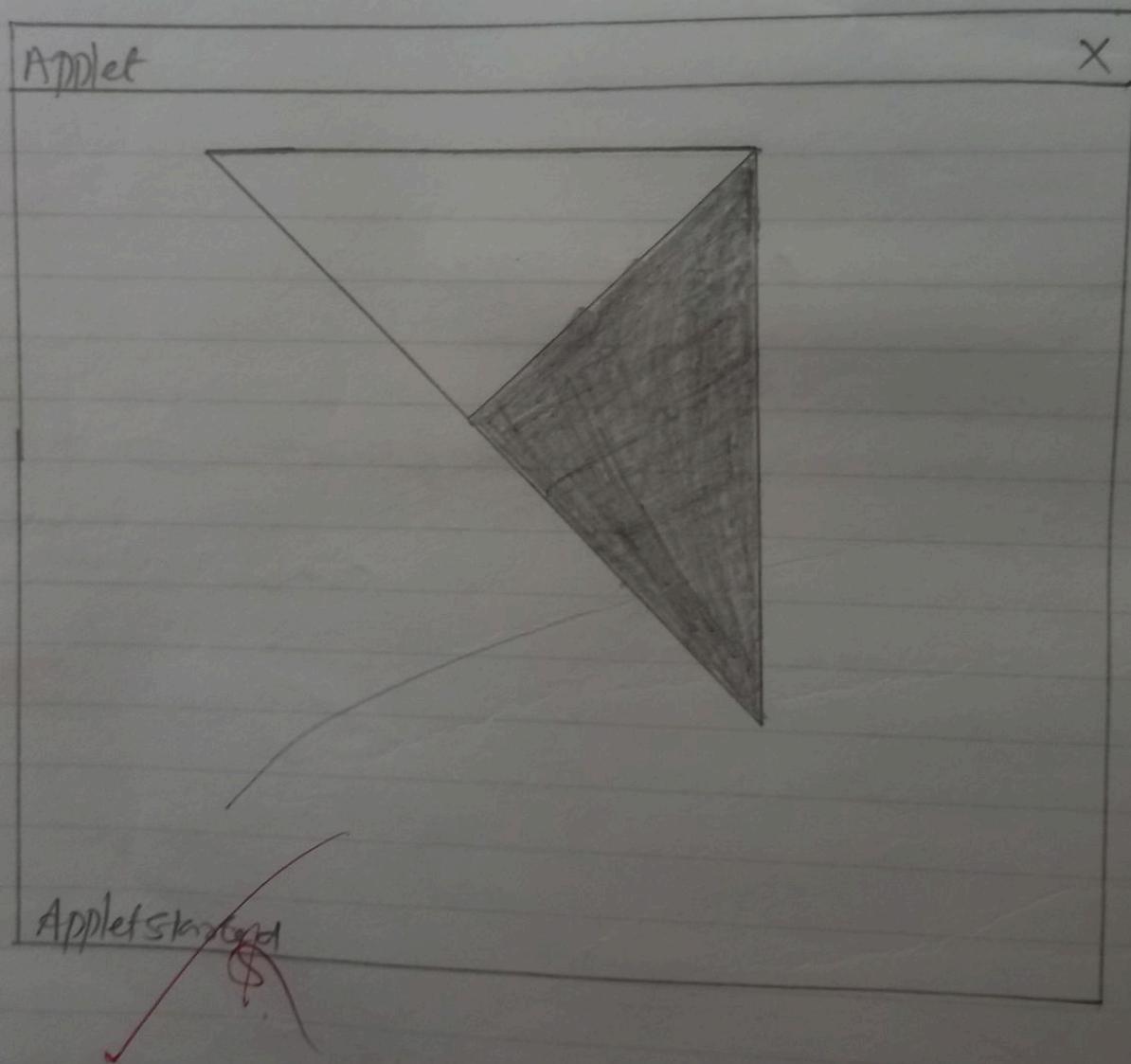
</html>

(6)

OUTPUT

D:\UIT>javac Poly.java

D:\UIT>appletviewer Poly.html



52

62

3X

PROGRAM NO: 21

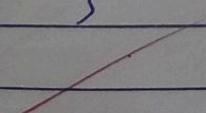
DRAWING POLYGONS

Write an applet program to draw polygons

program :

Note pad-1

```
import java.awt.*;
import java.applet.*;
public class poly extends Applet
{
    int x1[] = {20, 120, 220, 20};
    int x1[] = {20, 120, 20, 20};
    int n1 = 4;
    int x2[] = {120, 220, 220, 120};
    int y2[] = {120, 20, 220, 120};
    int n2 = 4;
    public void paint (Graphics g)
    {
        g.drawpolygon (x1, y1, n1);
        g.fillpolygon (x2, y2, n2);
    }
}
```



53

notepad - 2

32

```
<html>
<head>
<title> poly </title>
</head>
<body>
<applet code="poly.class" width=250 height=200>
</applet>
</body>
</html>
```