

Final

Interface

- অবশ্য method unimplemented থাকবে।
- interface এর object তৈরি করা যাবে না
- অবশ্য field final and static রয়ে।
- যেকোন object তৈরি করা যাবে না তবু field গুলোকে public করতে হবে।
- একটি প্লাট এর জন্যে interface এর ^{implements} ^{functions} বাস্তব নাওবে।
- যাকে implement করবো তা child হাবে আবার implement কু করবে unimplemented method গুলোকে পূরণ করতে হবে।
-

Ex:

```
interface A { }  
    double getArea();  
    void draw();  
}  
  
class B implements A {  
    private double w;  
    private double h;  
    public double getArea() {  
        return w * h;  
    }  
    public void draw() {  
        cout("Im B class");  
    }  
    public B(double w, double h) {  
        this.w = w;  
        this.h = h;  
    }  
}
```

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

```
        A in;
```

```
        B = obj1 = new B (4, 5);
```

```
        B obj2 = new B (4, 6);
```

```
        in = obj1;
```

```
        in.draw();
```

```
        in.getArea();
```

```
        in = obj2
```

```
        in.draw();
```

```
        in.getArea();
```

Output: Jim B class

Jim B class

24

IV) Partial Implementation

- যদি কোনো class এর মাৰ হোল্ডিং child গ complete না হয়ে থাকে, At-least ১০৫ uncomplete হওয়া পৈছ কথা - code error !
- সিলেক্ট অক্ষরে abstract declare কৰতে হবে,
- interface -কে implement কৰে তাহে implements ব্যব
abstract " " " " extends

```
interface Door {  
    void open();  
    void close();  
    void lock();  
}
```

```
abstract class LocklessDoor implements Door {  
    public void open() {  
        sout("open Door!");  
    }  
}
```

```
public void close() {  
    sout("Door closes!");  
}  
}
```

```
class FullyImplements extends LocklessDoor {  
    public void lock() {  
        sout("Door Locks!");  
    }  
}
```

```
public class PartialImplements {  
    public static void main(String[] args) {  
        Door dr;  
        LocklessDoor lr;  
        FullyImplements F = new FullyImplements();  
        dr = F;  
        dr.lock();  
  
        lr = F;  
        lr.lock();  
    }  
}
```

Output: Door Locks!
Door Locks!

4 Implementing an interface.

- ~~interface~~, interface द्वारा implements करते हैं।
मिलना extends करते हैं।
 - एक interface द्वारा कई classes implement करते हैं लेकिन extend करते हैं।
- class C extends A implements B, D

```
interface A {  
    void Door();  
}
```

```
interface B extends A {  
    void Lock();  
}
```

```
abstract class C implements B {  
    public void Lock() {  
        System.out.println("Door open, lock!");  
    }  
}
```

```
class D extends C {  
    public void door() {  
        sound("Door open!");  
    }  
}
```

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

- A a;
 - B b;
 - C d=new C();
 - D dr=new D();
- a = d;
a.Lock()

b = dr;
b.door();
b.Lock();

}

interface B
Door (parent)
Door (child)

Output: Door Lock!
Door open!
Door Lock!

▣ calendar

- new object - ক্লারি এক্সে প্রযুক্তি
- অন্যান্য method দ্বারা পরিচালিত মেথড হবে
- অক্টোবর
- month 0 = January 11 = December
- এবং calendar object import করবার পদ
- এবং import করে এই মেথড দ্বারা প্রাপ্ত রেস
- .getTime(); exact time প্রিন্ট করে আজাই
- set method দ্বারা এই object এর কিছু পরামর্শ মা করে ফেলতে হবে মনে।

▣ Time | set method

```
import java.util.Calendar;
```

```
public class Test {
```

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

```
        Calendar date = Calendar.getInstance();
```

```
        System.out.println(date.get(Calendar.YEAR)); // 2023
```

```
        date.set(Calendar.YEAR, 200)
```

```
        System.out.println(date.get(Calendar.YEAR)); // 200
```

```
sout( date.get(calendar.MINUTE)); // 08 current time  
date.set(calendar.MINUTE, 5);  
sout( date.get(calendar.MINUTE)); // 5  
  
sout( date.get(calendar.MONTH)); // current month  
date.set(calendar.MONTH, 62);  
sout( date.get(calendar.MONTH)); // 2
```

Mon NOV 13 17:22:51 BDT 2023
month month date Time country year
day

{ 17:22:51 BDT 2023 }

■ Add method

- current কামাদা করে নির্দেশ করে আবারু করে করে
- কামাদা করে করে করে করে করে করে করে করে

like current

current → Mon Nov 13 17:40:10 BDT 2023

```
import java.util.Calendar;
public class Test {
```

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

```
        Calendar Time = Calendar.getInstance();
```

```
        Time.add(Calendar.DAY_OF_MONTH, -2);
```

```
        Time.add(Calendar.MONTH, -4);
```

```
        Time.add(Calendar.YEAR, 4);
```

```
        Time.add(Calendar.MINUTE, 10);
```

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

|| Tue Jul 13 17:50:40 BDT 2027

}

田 getActualMaximum()

Time. getActualMaximum(Calendar. Field);

বেগুনী Field Field এর জায়গাম নিয়ে তা maximum
কর্তৃত করে নিয়ে তা current time কর

ex: Time. getActualMaximum(Calendar. MONTH);

|| 11 মাহ $11 + 1 = 12$ month possible

Time. getActualMinimum(Calendar. DAY_OF_MONTH)

11 30 \rightarrow নভেম্বর 6 সপ্তাহ 30 দিন - possible

①

②

田 Time zone

Timezone z = date. getTimezone();

Calendar object first time zone get
Sout(z); object কোথি কথি হচ্ছে।

|| Bangladesh Standard Time

৫) Exception → Run time error

→ code র ধরি error থাকলেও নির্বাচিত
error হয় সময়ে কোর্টে run করত না, কেন
কি এখনোর error or Identify কৰু ও solve
যথা করি exception

→ error কিছে try হয় প্রত্যে রাখতে হব,
→ কি ধরিয়ে error or catch করতে হয়,
যদে নিত হব,

- ① Arithmetic exception → A.E
 - ② Nullpointer exception → N.E
 - ③ ArrayIndexBound exception → A.I.B.E
 - ④ numberFormat exception → N.F.E
 - ⑤ IllegalArgument → I.A.E
- ৬) Try and catch block in Java

```
try {           → error
}
}           → error handling
catch (Exception e) { }
```

```
print ("Hello"); } Run 20% Output: Hell  
print ("20");  
int a = 2/0; → error.  
print (a); } Run 20% NR
```

```
print ("Hello"); ← Output: Hell  
print ("20"); ← 20  
try { ← wellcom  
    int a = 2/0;  
} ← yes
```

```
catch (Arithmetic e) {
```

```
    print ("welcome");
```

```
    print ("yes");
```

Exception

- Exception একটি object
- সূর্য code
- Exception মাঝে কোথাও তুল আবে ও ফিল
- এখন কোথাও error আবে পাবে? অফার কোম্পানি
- এখন কোথাও error আবে পাবে? ডেভেলপার
- এবং error কে try catch দিয়ে ধরবে।

① ArithmeticException → AE

মনে calculate করো ফর্স হয়ে error আবে
কোম্পানি

```
int a = 10/0;
```

```
try {
```

```
    int a = 10/0;
```

```
}
```

```
catch (ArithmeticeException e) {
```

```
    sout ("AE");
```

```
}
```

→ Try এর মধ্যে catch আছে

→ Try আছলে catch ব্যবহার করা হবে। must

iii) ArrayIndexOutOfBoundsException

Try {

 int s[] = new int[5];

 s[6] = 10;

 cout("Try Block");

} catch (ArrayIndexOutOfBoundsException e) {

 cout("Error"); // output: Error

}

N.B) Try block প্রয়োগ করে catch করা হবে।

 try block এর অন্তর্ভুক্ত কোডটি প্রয়োগ করা হবে।

 try block এর অন্তর্ভুক্ত কোডটি প্রয়োগ করা হবে।
 এখন কোডটি execute করা হবে।

 Catch এর অন্তর্ভুক্ত কোডটি run করা হবে।

- একটি try এবং তার এ্যান্ডিম কাচ মাঝে
 → একটি try এবং অর্থে এ্যান্ডিম Exception মাঝে
 এবং কাচ মাঝে এবং তার পাশে সেভিল
 মাঝে না।

```
try {
```

```
    int a = 10/0;
```

```
    int b[] = new int[3];
```

```
    b[5] = 10; X
```

```
    sout ("try block");
```

```
Catch ( ArithmeticException e ) {
```

```
    sout ("Exception Found");
```

```
}
```

```
Catch ( ArrayIndexOutOfBoundsException e ) {
```

```
    sout ("AIBE");
```

Output: Exception Found

Done

```
sout (" Done");
```

Exception Order

→ If exception goes catch करते तब

super exception या निचो या एवं sub या
उपरी या निचो रहते हैं।

1

try {

~~int~~ to int a = 10/0;

}

catch (ArithmeticeException e) {

} Sout('AE');

catch (ArrayIndexOutOfBoundsException e) {

 Sout("AIOBE");

catch (Exception e) {

 Sout("Any Exception");

}

[ग और अन्य Error fn]

4 Nested try - catch

```
int s[5] = new int [5];
```

```
try {
```

```
    sout ("Start try");
```

```
try {
```

```
    sout (" start change");
```

```
}
```

```
s[10] = 7;
```

```
}
```

```
sout ("After change");
```

```
}
```

```
catch (ArrayIndexOutOfBoundsException e) {
```

```
sout ("change catch");
```

```
}
```

```
sout ("main catch");
```

Output: Start try

```
sout ("After try catch");
```

start change

Output:

Change catch
After change
After try

if nested & each catch
inner try outer catch
G. understand it

句 throw

→ 異常 catch 張り立てる 亂す throw & 張る
→ 异常 catch 异常 抛出异常 catch 异常
抛出 异常 throw 异常 catch 异常 catch
□ 异常 异常 抛出 异常 catch 异常

try {

(Second try) too

~~int a = 10/0;~~ ; (A compile error)

? } try {

sout (" inner try"); ✓

int a = 10/0;

}

catch (Arithmeticeception e) {

sout ("throw"); ✓

throw e; ✓

}

sout (" inner . block"); ✗

catch (Arithmeticeception e) {

Output: inner try

throw

outer catch

}

sout (" outer catch"); ✓

[try method का call করতে পর স্বত্ত্বা catch করার
জন্য]

throws

→ মনি এমনো method কে call করা হবে তখন অ^২ method
যখন আপনি Exception throw করে আসলি কিংবা অ^১ call করে
আকে অপ্রাপ্য কোনো method এর প্রস্তাৱ দেখ

throw throws Exception name

public static getValues(int a) throws Exception {

if (a == 0)

{

 throw new Exception("Error");

}

else {

 return 100/a;

}

psvm (String [] args) {

try {

 getvalue(0);

Output : Found error

}

catch (Exception e) {

 cout ("Found error");
 cout (e.getMessage()); → স্বত্ত্বা দেখ

}

}

बाये ~~में~~ finally

- finally एक static के रूप में बना है,
- finally आपले प्रोग्राम बनाते ही execute होता है।

```
try {
```

```
    int a = 10/0;
```

```
} catch (ArithmeticException E) {
```

```
    sout("I am catch");
```

```
finally {
```

```
    sout("I am finally");
```

```
sout("After finally");
```

Output:

I am catch
I am finally
After finally

- ① ArithmeticException → AE
- ② NullpointerException → NE
- ③ ArrayIndexOutOfBoundsException → AIE
- ④ numberFormat~~Exception~~ → NFE
- ⑤ IllegalArgument → IAE

▣ Nested class

- nested class এর block বেঁধিতে বাইকে
পাওয়া যাবে না
- nested class এর immediate outer class এর
variable access করতে পারবে
- outer বিল্ডিং মোড়ে class খরে inner class
কে direct access করা যাবে না.

```
public class A () {
```

```
    private a = 10;
```

```
    private b = -10;
```

```
    class B () {
```

 pri

```
        public void U {
```

```
            sout (a)
```

```
            sout (b)
```

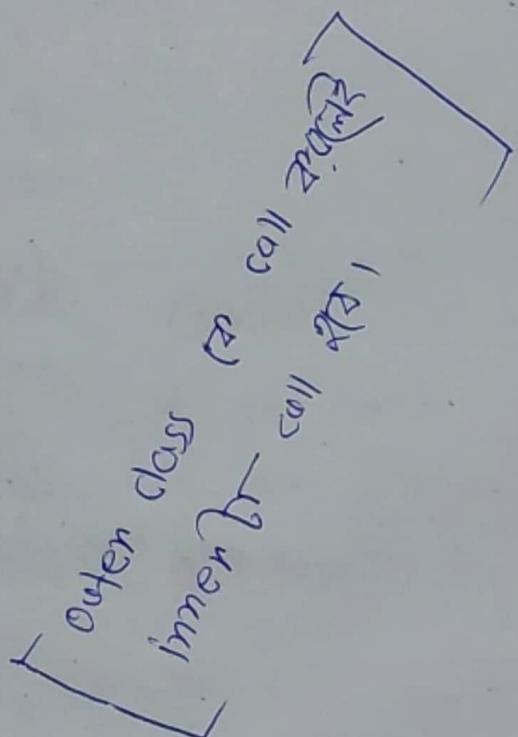
```
    }
```

```
}
```

```
psvm ( ) {
```

```
    A ob = new A ();      ✓ output: 10
```

```
    B ob = new B ();      X
```



4) Shadowing

```
public class shadowTest {  
    public int n=0;  
    class A {  
        public int n=1;  
        void method (int n){  
            cout ("n= "+n);  
            cout ("this.n = "+this.n);  
            cout ("shadow = "+shadowTest.this.n);  
        }  
    }  
}
```

Output : 23 n = 23

this.n = 1

shadow = 0

- block के लिए अप्रैर्यामिक नियम
- this.n अप्रैर्यामिक क्लास के लिए
- shadowTest.this.n → अप्रैर्यामिक बाहरी क्लास के लिए
बाहरी क्लास नाम

4) Anonymous class.

4) Local class

→ Nested class একটি class এর অন্তর্ভুক্ত কোড়ি
ব্যাখ্যা করুন।

→ Local class ও ঘরের method গুলি কোণ্ঠে কোণ্ঠে
ব্যাখ্যা করুন।

→ Execute কোণ্ঠে মনে রাখুন Local class গুলি কোণ্ঠে
কোণ্ঠে।

public int interface Interface {

 void method();
}

public class childA implements interface {

 public class B {

 public void method() {

 void method() {

 cout ("Local class

```
public interface A {  
    public void method();
```

{

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

```
    public class B implements A {
```

```
        public void method()
```

{

```
            System.out("method A");
```

{
}

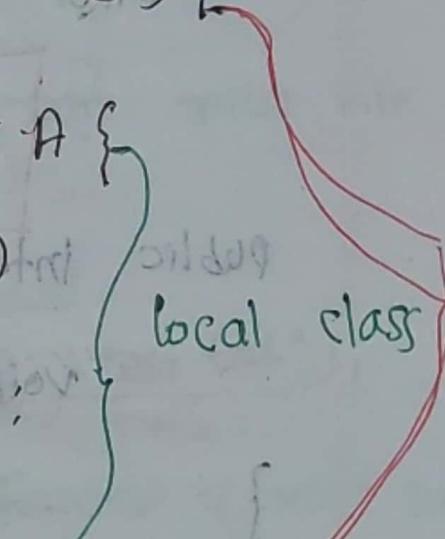
```
}  
B o = new B();
```

```
o. method();
```

{

main method

Output: method A



↳ Anonymous class

→ Anonymous class এর নাম থাকে না

→ object create করার সময় ক্লাস করার আজো method complete করতে হবে

```
A a = new A () {  
    implements  
};  
public interface A {  
    void method ();  
}  
public void static void main (String [] args) {  
    A a = new A () {  
        public void method () {  
            Sout ("Anonymous");  
        }  
    };  
}
```

V.V.I

```
import java.io.FileReader;
```

File Reader

- Object তৈরি করতে হবে। Object create করত
ক্ষেত্র সময় অন্মান যেখানে Read থাইলে ওটা করে আসে
১৩৫ Location এলে পিছু রে লিখে দেন (path name)
- আর ব্যাখ্যা হচ্ছে try-catch এর মতো কাব্য করে
খুঁজে না পাইলে খাতে কাজ করবে
- fileReader f = new FileReader("D:\\test.txt");
path
- FileReader এর object টা BufferedReader এ মিশ্র দিয়া।
BufferedReader bread = new BufferedReader();
- BufferedReader এর এই method দ্বারা IOException, name
Exception throws হয়ে।
- bread.readLine(); → গুরুতর Read থাবে
return type String
- f.close(); } close না করলে ঘূর্ণ প্রয়োগ করে File
or access মিল না পাবে।

text

```
Hi  
I am Rakib  
ID: 0112310308
```

```
public class FileRead {
```

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

```
        try {
```

```
            FileReader f = new FileReader ("D:\\text.txt");
```

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

```
            BufferedReader bRead = new BufferedReader (f);
```

```
            ("text.txt"))
```

```
bRead.read
```

```
int file = bRead.read();
```

```
String s = bRead.readLine();
```

```
if (file >= 0) {
```

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

```
}
```

```
catch (FileNotFoundException e) {
```

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

```
catch (IOException e) {
```

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

// Output: Hi

for entire file read

[code same as previous page]

while(true){}

tent

Hi
I am Rakib
ID: 0112310308
Fi-Amanullah

String s = ~~input~~ bRead.readLine();

If (s == null)

{
break

}

sout(s);

// Output: = Hi

I am Rakib

ID: 0112310308

Fi-Amanullah

}

((103103) f1123103.readLine() = null fail

}

田 Read এবং sum মেরে print করো ২টা

[code পাঠ্য কুই]

while (true) {

String s = bRead.readLine();

if (s == null)

{
break;

}

String [] arr = s.split(" ");

int n1 = Integer.parseInt(arr[0]);

int n2 = Integer.parseInt(arr[1]);

int sum = n1 + n2;

sout(sum);

}

23	23
8	5
6	7
3	8

Output :
46
13
13
11

import javax.swing.

Path input from user

→ JFileChooser এর object তার করো ৱৰ্ত।

→ object.showOpenDialog(null);

File return কৰ্বে

→ showOpenDialog int return যুক্ত ৩ type এর।

থিনি returning JFileChooser.APPROVE_OPTION ২৫ তারিখে

থিনি কৰ্বে

psvm (String args[]){

JFileChooser fc = new JFileChooser();

int response = fc.showOpenDialog(null);

if (response == JFileChooser.APPROVE_OPTION)

{

File f = fc.getSelectedFile();

try {

fileReader fr = new FileReader(f);

BufferedReader bRead = new BufferedReader(fr)

}

Import java.nio.file.DirectoryStream

- Files
- Path
- Paths

■ Path Tester

→ আমাৰ যেই path তুলি Location টে দেখুলো পি
আকে আমাদেৱ (Laptop/pc) টে আছে বিনা কোনো চেক
বুথে।

→ Path.toAbsolutePath(); → original path return কো

→ Files.isDirectory(path); → এই path টি ফলো কো
টি এখন Folder বিনা কো অন্তৰ মাছে।

→ Files.getLastModifiedTime(path); → লাস্ট কো
Update কো কো তাৰ সময় ফলো

→ Files.size(path); → এইচি কো অন্তৰ আছে কো
path কো সফল কো জ্ঞা কো।

→ FileObject.exists(); true/false ফলো কো file
আকে কো কো আছে।

→ File.list() → এই Directory কো গুড়ো কো
মাছে কো [String[] s = File.list();] string

```
import java.util.Scanner;  
import java.io.File;
```

Another way to Read

```
public class Read {  
    public static void main(String[] args) {  
        try {  
            File f = new File("Location/Student.txt");  
            Scanner sc = new Scanner(f);  
            while (sc.hasNext()) {  
                String id = sc.next(); }  
                String name = sc.next(); }  
                System.out.println("ID: " + id + " Name: " + name);  
            }  
            sc.close();  
        }  
        catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
    }
```

[~~\$~~ Stream use এখা এই binary file G- মিয় আর্স]

is.skip(8) → এখন 8 byte skip করবে।

Another way for read

→ InputStream এবং Object তারি করতে হবে

InputStream is = new FileInputStream("file.txt"); Object filo =

is.available(); → এতে byte head ফর বাবি আভ হত
বলা নিয়ে।

is.read() → এই byte read কোর হয়, তাৰ ASCII value
return কৰবে।

(char); is.read() → byte এ char কৰ কৰে কৰ্মসূচি

(is.available() > 0) → আর byte read কৰবে।

public class A {

 public static void main(String[] args) { File fi = new File("output.txt");

 try { Input Stream is = new FileInputStream("output.txt"); fi

 while (is.available() > 0)

 {

 sout((char)is.read());

 }

 } catch (Exception e) {

 sout(e.getMessage());

 }

File write

- according to file Read
- FileWriter এর object তৈরি করতে হবে
- যাইহো এমন try catch দ্বাৰা মনে রাখিব
- ```
FileWriter fw=new FileWriter("D:\\Output.txt");
```

 path
- FileWriter এর একটি object দ্বাৰা BufferedWriter দ্বাৰা দিব।

```
BufferedWriter bwrite = new BufferedWriter(fw);
```

→ BufferedWriter এর method দ্বারা IOException হবে

```
bwrite.write("Hello");
```

 String write হবে।

```
fw.close();
```

 এইটি সেই কোড।

```
bwrite.close();
```

 এইটি সেই কোড।

```
;()
```

 এইটি সেই কোড।

```
} ()
```

 এইটি সেই কোড।

```
()
```

 এইটি সেই কোড।

```
public class write {
 public static void main (String [] args) {
 try {
 FileWriter fw = new FileWriter ("output.txt");
 BufferedWriter bwrite = new BufferedWriter (fw);
 bwrite.write ("CSE life is very painful");
 fw.close();
 bwrite.close();
 } catch (IOException e) {
 System.out.println (e.getMessage());
 }
 }
}
```

CSE life is very painful

OUTPUT

import. java.io.FileOutputStream

Another way to write

- FileOutputStream एवं object द्वारा लिखा जाता है

File f = new File("Output.txt");

FileOutputStream Fos = new FileOutputStream(f);

→ byte ताराएँ इनपुट ड्रॉफ

→ Fos.write(byte);

String s = "CSE life ---";

Fos.write(s.getBytes());

→ Fos.close();

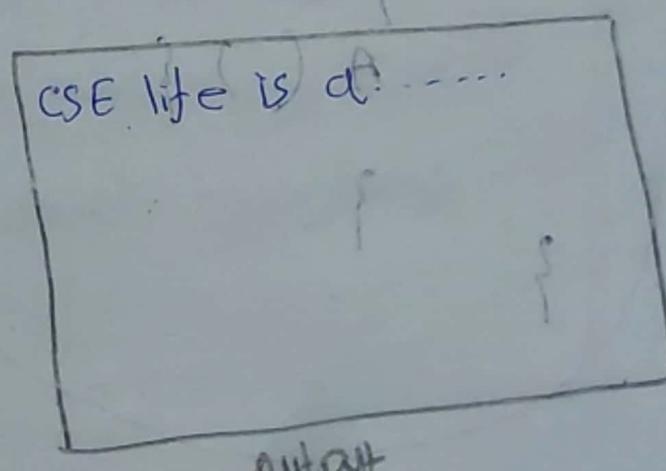
→ Fos.flush();

→ byte buffer = toInsert.getBytes();

इसमें array तरीके byte को convert करते

→ ऑफ

```
public class OutputStream {
 PSVM (String t) {
 try {
 File f = new File ("output.txt");
 FileOutputStream fos = new FileOutputStream (f);
 String s = "CSE life is a---";
 fos.write (s.getBytes ());
 fos.flush ();
 fos.close ();
 } catch (IOException e) {
 System.out.println (e.getMessage ());
 }
 }
}
```



import java.io.Serializable

object copy and save

→ Serializable को implement करा सकते हैं और पार्स

→ save के Binary भाष्यक

→ ऐसे class को write कर सकते हैं जो अन्य class

द्वारा object के रूप में लिया जाएगा जो serializable

public class A implements Serializable {

private int n;

private int y;

A (int n, int y) {

this.n = n;

this.y = y;

A () {

}

}

class A Serializable {  
}  
class B Serializable {  
A ob = new A();  
}

[Object write test]

psvm (string [] args) {

A m1 = new A (100, 200);

try {

~~ObjectOutputStream oos = new~~

~~FileOutputStream fs = new FileOutputStream ("output.dat")~~

~~ObjectOutputStream oos = new ObjectOutputStream (fs)~~

oos.writeObject(m1);

oos.close();

} catch (Exception e) {

sout (e.getMessage());

}

}

## [Object read করা]

```
psvm (String [] args) {
 try {
 FileInputStream fs = new FileInputStream("OUTPUT.OBJ");
 ObjectInputStream ois = new ObjectInputStream(fs);

 A m2 = (A) ois.readObject();
 // এইটি কোডের মধ্যে Object.readObject টাইপ
 // read করে মাত্র নকুল রাখা হয়েছে।
 // এইটি কোডের মধ্যে type cast করা হয়েছে।
 // এখন m2 ছাড়া class A এর মোট ফাংশন
 // নাই।
 } catch (Exception e) {
 System.out.println(e.getMessage());
 }
}
```

## Wrappers and Collections

→

Wrapper

Character

Boolean

Byte

Integer

Short

Long

Float

Double

Primitive

char

bool

byte

int

short

long

float

double

→ Wrapper object তৈরি করো

→ Primitive value thru Direct

primitive

int a = 10;

double b = 20.0;

wrapper

Integer ob = new Integer(a);

Double ob = new Double(b);

boxing

double d = ob.doubleValue();

int i = ob.intValue();

unboxing

## collections

- Types of Array
- store এবং মাপ করা যাবে, manipulation করা যাবে,
- collections অনেক ছুলা object মিহি কাজ করে,
- import java.util.collections;
- ArrayList কি sort করা যাবে
- collections.sort (List কি নাম);  
কি সিসি মধ্যে কোথা data কো দেখা যাবে কি নামে  
sort করে (primitive, user defined & কোথা)

myList

|   |
|---|
| 9 |
| 3 |
| 6 |
| 4 |

→ collections.sort (myList)

|   |
|---|
| 3 |
| 4 |
| 6 |
| 9 |

(ArrayList) list = b

(Collections) sort = i

- ① add ( object ): → sequence অন্তর্ভুক্ত করে  $[0, 1, 2, 3]$   
index
- ② clear(): → List কে মুছে দিব।
- ③ isEmpty(); → List এ কিছু ঘোচ নির্বাচিত নয়।
- ④ remove ( object ): → particular object remove করে।
- ⑤ addAll ( ~~list~~ listname ): → একই list এর element এবং কিছু করা।
- ⑥ size(): → List এর size জানা আবব।
- ⑦ removeAll ( ~~list~~ list name ): → মুছে দেওয়ার পরে clear()।
- ⑧ add ( index, object ): → add করার স্থান index বলে দেওয়া।
- ⑨ get ( index ): → কোনো index এর object get
- ⑩ lastIndexof ( list ): → List এর last index বিরে দেওয়া।
- ⑪ indexOf ( object ): → object এর index খুঁজে।
- ⑫ remove ( index ): → particular index এর object কে remove করে।

Type for call

Listname. method();

ArrayList< Integer > list = new ArrayList();

list.add(); → Integer এর একটি কিছু add করবেন।  
list.get();

ArrayList A = new ArrayList();

→ Data type रूप से नहीं लिये गए तो Data type

इस add का रूप नहीं होता।

A.add("Rakib"); → inferred or default value

A.add(12); → inferred value

A.add(308); → inferred value

A.add(308.0); → "3"

जब वेब्सीट पर इनपुट करते हैं तो (वेब्सीट) का

इनपुट को डिटेक्ट करते हैं तो (वेब्सीट) को डिटेक्ट करते हैं

जब वेब्सीट पर डायड देते हैं तो (वेब्सीट) को डायड देते हैं

जब वेब्सीट पर डायड देते हैं तो (वेब्सीट) को डायड देते हैं

जब वेब्सीट पर डायड देते हैं तो (वेब्सीट) को डायड देते हैं

जब वेब्सीट पर डायड देते हैं तो (वेब्सीट) को डायड देते हैं

जब वेब्सीट पर डायड देते हैं तो (वेब्सीट) को डायड देते हैं

जब वेब्सीट पर डायड देते हैं तो (वेब्सीट) को डायड देते हैं

जब वेब्सीट पर डायड देते हैं तो (वेब्सीट) को डायड देते हैं

जब वेब्सीट पर डायड देते हैं तो (वेब्सीट) को डायड देते हैं

जब वेब्सीट पर डायड देते हैं तो (वेब्सीट) को डायड देते हैं

जब वेब्सीट पर डायड देते हैं तो (वेब्सीट) को डायड देते हैं

## ArrayList sort

ArrayList < Integer> new A = new ArrayList();

A.add(5);

A.add(3);

A.add(39);

A.add(2);

| A  |
|----|
| 5  |
| 3  |
| 39 |
| 2  |

Collections.sort(A);

| A  |
|----|
| 2  |
| 3  |
| 5  |
| 39 |

Ascending  
order

ArrayList < String> B = new ArrayList();

B.add("Rahib");

B.add("Fardin");

B.add("Tamim");

B.add("Yasin");

|        |
|--------|
| Rahib  |
| Fardin |
| Tamim  |
| Yasin  |

Collections.sort(B)

| B      |
|--------|
| Fardin |
| Rahib  |
| Tamim  |
| Yasin  |

Alphabetically

or Lexicographical

In my class

```

public class Myclass implements Comparable {
 int a;
 int b;
 Myclass() {
 A

	A	A
S		
E		
C		
R		

 myclass (int a, int b) {
 this.a = a;
 this.b = b;
 }
 public int compareTo (Myclass o) {
 if (this.n == o.n)
 return (this.y - o.y);
 else
 return (this.n - o.n);
 }
 }
}

```

```
public class Test {
 public static void main(String[] args) {
 ArrayList<Myclass> list = new ArrayList<>();
```

A = new Myclass(10, 20);

B = new Myclass(12, 30);

C = new Myclass(10, 18);

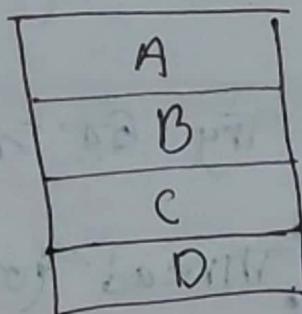
D = new Myclass(18, 35);

list.add(A);

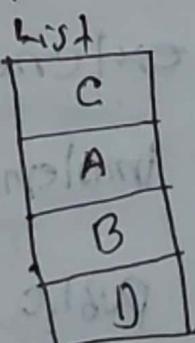
list.add(B);

list.add(C);

list.add(D);



Collections.sort(list);



$x = 10, y = 18$   
 $x = 10, y = 20$   
 $x = 12, y = 30$   
 $x = 18, y = 35$

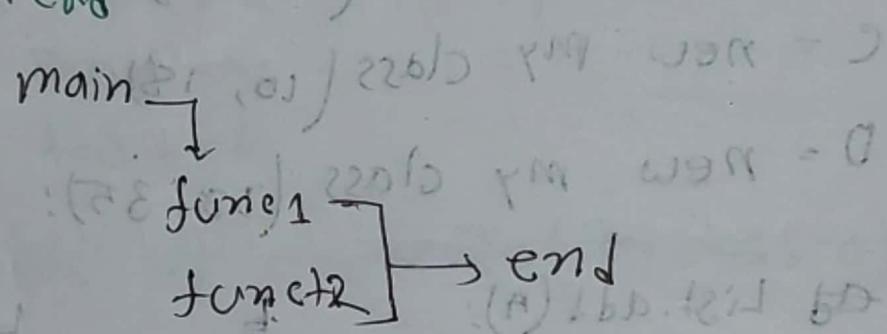
## Thread

→ এনের গুলি operation করা হয়।  
thread আর নাই (২৫০ ms র প্রতি) হচ্ছে

⊗ Without threading

main → func1() → func2() → End = A

⊗ With thread



⊗ Try কর সক্ষম হবে 'thread' এর

⊗ Thread করি বলো জন

① Extending Thread class

② Implementing the Runnable class

public void run();

Can not change signature.

Creating thread by extending Thread class

```
class Thread1 extends Thread {
 public void run() {
 for (int i=0; i<10; i++)
 {
 System.out.println(Thread1 + i);
 }
 }
}
```

for loop runs in run method  
start() calls run method  
in main class

```
public Thread2 extends Thread {
```

```
 public void run() {
 for (int i=0; i<10; i++)
 {
 System.out.println("Second thread is running");
 System.out.println("Thread2 " + i);
 }
 }
}
```

```
class Main {
```

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

```
 Thread1 T1 = new Thread1();
```

```
 Thread2 T2 = new Thread2();
```

```

try {
 T1.start();
 T2.start();
} catch (Exception e) {
 sout (e.getMessage());
}

```

ফিল্ড - T<sub>1</sub> এ output buffer  
 ফিল্ড - T<sub>2</sub> এ output buffer  
 Thread 1  
 Thread 2  
 Thread 2 2  
 Thread 2 3  
 Thread 1 2  
 Thread 1 3

```

try {
 t1.start();
 t1.join();
 T2.start();
 T2.join();
} catch (Exception e) {
 sout (e.getMessage());
}

```

Join() মধ্যে এখন রাখ  
 রান্ন মধ্যে thread এখন রাখ  
 আরেক ক্ষুণ্ণ রাখতে পাই  
 cursor must complete  
 রান্ন মধ্যে thread  
 ৬ মাত্রা, এখন রাখ  
 Thread 1 1  
 Thread 2 1  
 Thread 2 2  
 Thread 2 3

→ এই thread এর join() মাধ্যমে আপনার প্রোগ্রাম  
সমাপ্ত নয় কিন্তু complete না হওয়ে থেকে thread এ  
থাকে না।

Thread.sleep(300) → millisecond

এই Thread execute করে 300 millisecond R  
পর অন্য Thread এ চালে।  $\{1000 \text{ ms} = 1 \text{ s}\}$

Thread.join(300) → 300 ms অন্তরে বাস্তবে Thread টি  
কাশ করে ছাড়া,

Thread.setName(" "); Thread এর নাম কিটি করা হবে  
Thread.getName(); Thread এর নাম কানেক্ট করা

Thread.join(); এই Thread কে না রেখা স্বত্ত্ব  
অন্য Thread কে সমাপ্ত করে  
অন্য execute করে না

Thread.currentThread(); বর্তমান মেই Thread কাজ করছে  
কিন্তু কুশল,



## GUI

- Graphic user interface = Gui
- এই কোড পপ-আপ ওপেন করে।  
মুন্ত প্রোগ্রাম করে boring বা game/  
software কোড করতে হবে।
- আজ এই package import করে।  
import java.util.swing.\* এই method আছে।

ShowConfirmDialog() → yes/no/cancel

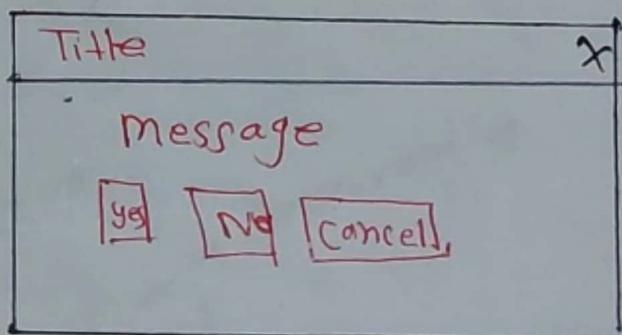
ShowInputDialog() → input নিষ্ঠা কর; অথবা  
string ভিত্তি আলাদা নিষ্ঠা

ShowMessageDialog() → কোন message দাখাই  
যাবে অথবা button

### Confirm

Syntax: JOptionPane.showConfirmDialog(null, "Message",  
"Title", JOptionPane.INFORMATION\_MESSAGE);

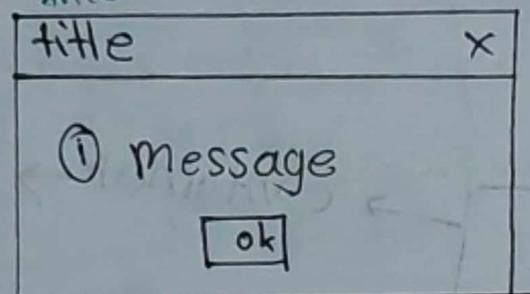
এই icon এখন জন্ম।  
অধাৰ-আৱা-অনু icon।  
দিত সাধাৰণ।



```
import java.awt.swing.JOptionPane;
```

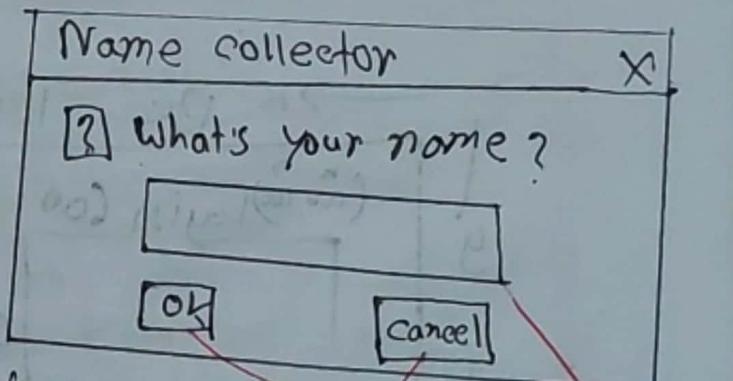
## Message

Syntax: JOptionPane.showMessageDialog(null, "message",  
"title", JOptionPane.INFORMATION\_MESSAGE),  
icon



## Input

Syntax: JOptionPane.showInputDialog(null, "What's your  
name?", "Name collector", JOptionPane.QUESTION  
-MESSAGE);



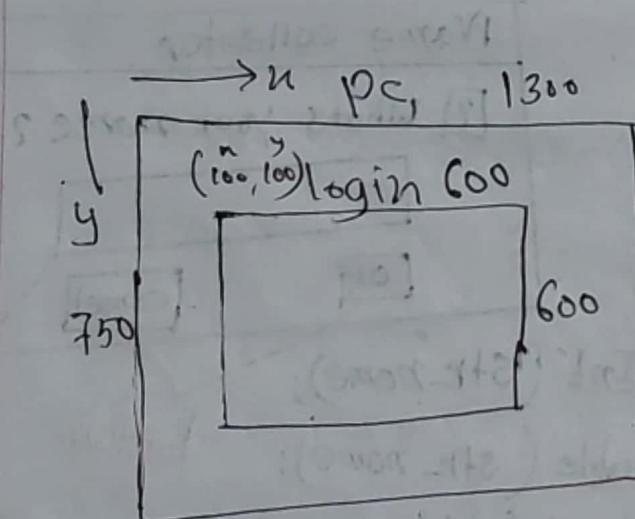
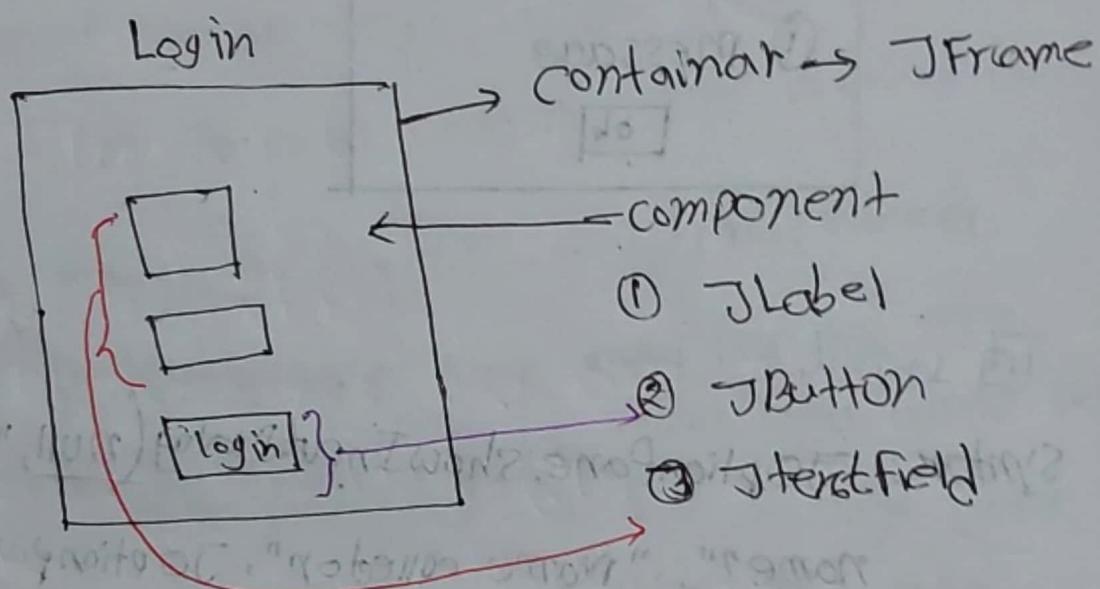
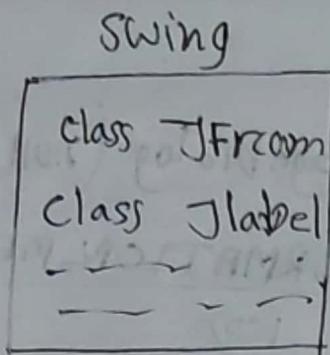
```
int a = Integer.parseInt(str_name);
```

```
double d = Double.parseDouble(str_name);
```

```
float f = Float.parseFloat(str_name);
```

```
boolean b = Boolean.parseBoolean(str_name);
```

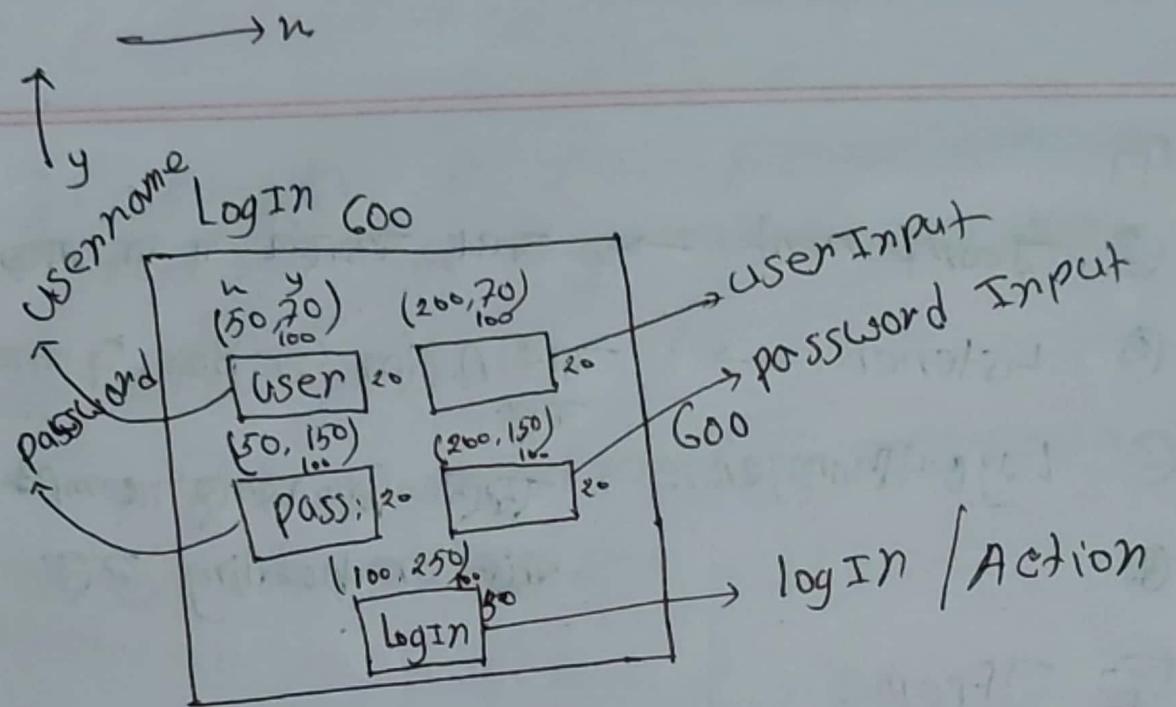
GUI → package swing



setBounds(n, y, w, h)

① Location → setLocation(n, y)

② Size → setSize(w, h)



Layout (null) → മാനുഖിക (അല്ലെങ്കിൽ ഫോറ്മേറ്റ്)

അല്ലെങ്കിൽ സൗജ്യം, താഴെയായി പറയുന്നത്.

public class GUI {

    JFrame j = new JFrame();

    public GUI {

        j.setLocation(100, 100);

        j.setSize(600, 600);

        j.setVisible(true);

}

## ① JComponent

→ अलग प्रकार की वा लाई

② Listener → याथ Action द्वारा Basis से करें

③ LayoutManager → इसके द्वारा बनाए गए component  
वाले वा handling करें

## ④ JFrame

① setVisible (true)

Parameter true किसी भी तरह वा नहीं करें

② setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);

Run close करने वाले Automatic close हो जाएं

③ setSize (width, height);

frame के आकार को वा इस वा ऐसे बदलो

④ setLocation (x, y);

monitor द्वारा position के वाले वा ऐसे बदलो।

⑤ setLocationRelativeTo (Null);

नाहीं Auto Location हो वा centre हो

⑥ setBounds (width, height, x, y);

⑦ get; ⑧ number द्वारा मात्र बदलते ही नहीं होंगे

⑦ setTitle (" " );

Frame এর title ফিরে আবারু রেখা করতে হবে।

⑧ setResizable (False)

মানুষ frame কে স্লিপ-ড্র ও mouse মাইসের  
চাপে রেখা করতে বাব্বন্ত। } NOT

public class FrameDemo extends JFrame {

psvm (String [] args) {

FrameDemo f = new FrameDemo ();

f.setVisible (true);

f.setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);

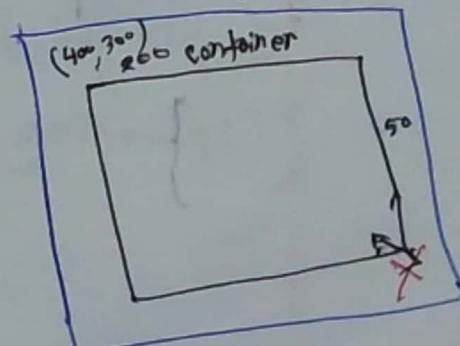
f.setBounds (200, 50, 400, 300);

f.setTitle ("container");

f.setResizable (false);

ফিলে-ই ঘোষণা করা না - Frame

ওর Resize করতে না।



## || Another way

```
public
import javax.swing.JFrame;
public class Test extends JFrame {
 Test () {
 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 setBounds(500, 170, 600, 500);
 setTitle("title");
 setResizable(false);
 }
 public static void main (String [] args) {
 Test f = new Test ();
 f.setVisible(true);
 }
}
```

## Icon change of JFrame

```
import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.GridLayout;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.WindowConstants;
```

```
public class Test extends JFrame {
 Test() {
 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 setBounds(100, 100, 600, 500);
 setTitle("calculator");
 setResizable(false);
 iconAdd();
 }
 public void iconAdd() {
 ImageIcon icon = new ImageIcon(getClass().getResource("img.png"));
 this.setIconImage(icon.getImage());
 }
 public static void main(String[] args) {
 Test f = new Test();
 f.setVisible(true);
 }
}
```

```
import java.awt.Container;
import java.awt.Color;
```

## Java JFrame background color

→ constructor ഏ ഫിൽ Container type യും  
എൻ്റെ variable സൈറ്റ് കുറഞ്ഞതാണ്  
variable.setBackground (color.type) നി ദില്ലേ ഒരു

```
import java.awt.Container;
```

ഓരോ ടോളി മാർക്കു കുറഞ്ഞ  
ഒരു color എംബു modify

```
Test () { }
```

```
Container c = this.getContentPane();
c.setBackground (Color.BLACK);
```

```
}
```

```
}
```

```
{ }
```

```
{ }
```

```
{ }
```

```
{ }
```

import java.awt.swing.JLabel

## 困 JLabel

Container এর Layout (null) দিতে যাবে  
এবং

① setText();

② setBounds();

→ JLabel হাতে text and image add করা যাবে  
→ JLabel এর object create করে স্থির object  
করে modify করে container এর object এ add  
করতে পারে।

previous এর মাধ্যমে মাত্র নথি constructor  
কি change করে।

Test() {

component();

}

public void component() {

Container c = this.getContentPane();  
c.setLayout(null);

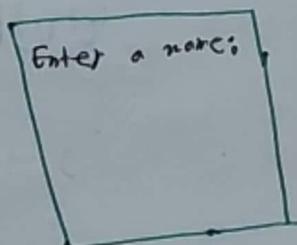
JLabel L = new JLabel();

L.setBounds(x: 50, y: 20, w: 150, h: 50);

L.setText("Enter a name");

c.add(L);

}



## • multiple JLabel set

→ setBounds ~~करते~~ G value आवाया नहीं  
आजे एके Label बदला करा दिले ते किसीका  
कम नहीं layout size किये रखे।  
(मध्ये आपले इस उ खाले Label) जूनीचे height  
जूनीचे count घावले रखा।

Test () {

component ();

}

public void component () {

Container c = new this.getContentPane();

c.setLayout(null);

JLabel L1 = new ("Enter name:");

L1.setBounds (x: 50, y: 20, w: 150, h: 50);

c.add (L1);

JLabel L2 = new ("pass: ");

L2.setBounds (50, 80, 150, 50);

if ("").length > 10 {

c.add (L2);

}

Enter name:  
Pass:

## Front style and size

→ Font type ଏହି କେବଳ variable ମାତ୍ର ହୁଏ

→ यहाँ use करते ही object create करेंगे

```
254 f=new Font("Arial", Font.BOLD, H);
 name color size
```

→ userLabel.setFont(f)

Label variable

```
import java.awt.Font;
```

private Font font;

```
Test() {
```

## compon e

```
public void component () {
```

```
f = new Font("Arial", Font.BOLD, 14);
```

L1.setfont(f);

21.09.12) 1409070708 • 1

(holly mello) ~~bump~~ the i

```
L2.setFont(f);
```

JLabel Foreground and background color

- JLabel এর text color ব্যবহার করে পরিবর্তন করা যাব।
- JLabel এর background color change করার জন্য আজো L1.setOpaque(true); ব্যবহার কর

code মাঝেও দেখু

Container এ এক ফলাফল মাঝে JLabel এর  
বাস্তব ফর্ম দেখতে হবে।

Test () {

    component(); } // main()

}

public void component () {

    L1.setOpaque(true);

    L1.setForeground(Color.RED);

    L1.setBackground(Color.YELLOW);

} e.add(L1);

}

এখানে একটি Label এ ~~কোনো বা লিঙ্গ~~  
বাদের কাছে ~~কোনো~~ | return করে  
string

set Tool Tips | getText();

→ এই Label-এ প্রকথনের Label এর কাছে  
cursor নিলে একটি information আঘাত।

Test () {  
 component();

}

public void component () {

L1.setToolTipsText (" prompt message");

String s = L1.getText();

cout(s) || Enter name

String s1 = L1.getToolTipText();

sout(s1) || prompt message

ImageIcon import করুন ২৫।

- By image image set in a JLabel
- JLabel টেক্সট বর্ণনা করে রেখে এবং একটি প্রতিক্রিয়া মাধ্যম নির্দেশ করে রেখে।
- ImageIcon এর object create করে রেখে আর মাঝের মধ্যে মিল করে রেখে।

Text () {

component();

public void component () {

ImageIcon img = new ImageIcon (getClass().  
getResource ("img.png"))

L3 = new JLabel (img);

L3.setBounds (50, 30, w: img.getIconwidth(),

h: img.getIconheight ());

c.add (L3);

(image এর W, H)

}

## 田 JTextfield

- আপনা কিন্তু input করতে
- JTextfield এর object এর বাস্তব রূপ
- object এর modify করাবে মাত্র container  
এর object এর add করে নিজে

```
Tent () {
 component();
}
public void component () {
 Container cn = this.getContentPane();
 JTextField jt = new JTextField ();
 jt.setBounds (x: 50, y: 100, w: 150, h: 30);
 jt.setBackground (color. WHITE);
 cn.add (jt);
}
```

## ④ Align text field

→ JTextField এর কোন সীরিজ মধ্যে ইন্সেন্স নিব  
or এটি এব্যাপ্তি !

→ JTextField গুরুত্ব ছাড়া কল মিহু কর

f1.setHorizontalAlignment ( JTextField.CENTER );

→ container এ add করার আগে রাখো কর

Text () {

component();

}

public void component () {

contenten cn = this.getContentPane();

JTextField f1 = new JTextField();

f1.setHorizontalAlignment ( JTextField.CENTER );

cn.add (f1);

}

## Action Listener

→ ऐसा framed input के OBJECT पर  
किसी कि घटना तो कहि Action Listener पर  
कानून

### ActionListener

```
public void actionPerformed();
addActionListener();
```

```
Test {
```

```
component();
```

```
}
```

```
public void component() {
```

```
c = this.getContentPane();
```

```
c.setLayout(null);
```

```
c.setBackground(Color.ORANGE);
```

```
JTextField j = new JTextField();
j.setBounds(50, 50, 150, 50);
c.add(j);
```

```
j.addActionListener(new ActionListener(){
 public void actionPerformed(ActionEvent)
 {
 String s = j.getText();
 if (!s.isEmpty()) {
 JOptionPane.showMessageDialog(null,
 "J = " + s);
 }
 else {
 JOptionPane.showMessageDialog(null,
 "you didn't enter anything");
 }
 }
});
```

↳ Listener close      (Output - JOptionPane)  
}      Component close      (Input - Print)

## ActionListener

→ getSource() में Button एवं object का नाम जैसा कि Button click का,

→ इसी फलाने से ActionEvent एवं object का नाम लिया जाता है।

PW

```
i() targetep.t = e.getSource();
public class ActionListener extends JFrame {
```

```
 ActionListener() { }
```

```
 component();
```

```
}
```

```
public void component() {
```

```
 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
 setTitle("calculator");
```

```
 setResizable(true);
```

```
 setBounds(100, 50, 500, 400);
```

```
container cn = this.getContentPane();
```

```
JTextField f1 = new JTextField();
```

```
f1.setBounds(50, 50, 150, 50);
```

```
c.add(f1);
```

```
JTextField f2 = new JTextField();
```

```
f2.setBounds(50, 110, 150, 50);
```

```
c.add(f2);
```

```
Handler han = new Handler();
```

```
f1.addActionListener(han);
```

```
f2.addActionListener(han);
```

```
} // component Method end
```

```
class Handler implements ActionListener {
 public void actionPerformed(ActionEvent e) {
 if (e.getSource() == f1) {
 String s1 = f1.getText();
 String s2 = f2.getText();
 int a1 = Integer.parseInt(s1);
 int a2 = Integer.parseInt(s2);
 int result = a1 + a2;
 Jlabel r = new Jlabel();
 r.setBounds(100, 150, 150, 50);
 r.setText(" " + result);
 r.setSize(150, 50);
 r.setLocation(100, 150);
 r.setVisible(true);
 } else {
 }
 }
}
```

} || closed ActionPerformed

} || closed Handler class

```
import java.awt.swing.JPasswordField;
```

## Java JPasswordField

- \* JPasswordField();
- \* setEchochar('\*') → Hidden character add করে
- JPasswordField এর object পরি রাখো করে।

```
constructor() {
```

```
 component();
```

```
}
```

```
public void } component() {
```

```
JField
```

```
JPasswordField jt = new JPasswordField();
```

```
jt.setBounds(50, 20, 150, 50);
```

```
jt.setEchochar('*'); → এই প্লাট পাসওয়ার্ড
```

```
c.add(jt);
```

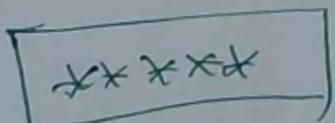
type করতে নাব

মুক্ত চর পৰি

পৰি শো কৰিব

hidden হিসেবে

```
}
```



## ▣ JButton()

→ object create করতে হবে

→

```
constructor () {
```

```
 component();
```

```
}
```

```
public void component () {
```

```
 ch = this.getContentPane();
```

```
 JButton bt = new JButton("Button");
```

```
 bt.setBounds(100, 50, 100, 50);
```

```
 ch.add(bt);
```

```
}
```

Button name অন্তিম  
করে আনতে অসম্ভব

bt.setText(" ")

④ setcursor() / cursor

→ object create new cursor or

constructor () {

component();

}

public void component () {

cursor cr = new cursor(cursor.HAND -  
cursor);

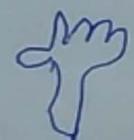
JButton bt = new JButton("Create");

bt.setCursor(cr);

}

cursor ga-name at ?

the name



Hand

## 4) image add in a Button

→ ImageIcon object create

→ Then Button object is add this

```
constructor() {
```

```
 component();
```

```
}
```

```
public void component() {
```

```
 ImageIcon img = new ImageIcon(getClass().
```

```
 ("login"));
```

```
 JButton bt = new JButton(img);
```

Image name

```
}
```

Image type

Image object add

## ActionListener add on JButton

```
constructor () {
 component ();
}

public void component () {
 c = this.getContentPane();
 JTextField tf = new JTextField ("welcome");
 c.add (tf);
 JButton bt = new JButton ("clear");
 c.add (bt);
 bt.addActionListener (new ActionListener () {
 public void actionPerformed (ActionEvent e)
 {
 tf.setText ("");
 }
 });
}
```

মন্তব্যঃ bt.Button click করে তখন tf  
clear এর ফল।

## ④ Login Frame

```
Constructor () {
 component();
}

public void component () {
 Container c = this.getContentPane();
 JLabel l1 = new JLabel ("User name");
 c.add (l1);
 JTextField f1 = new JTextField ();
 c.add (f1);
 JLabel l2 = new JLabel ("password ");
 c.add (l2);
 JTextField f2 = new JTextField ();
 c.add (f2);

 JPasswordField pf = new JPasswordField ();
 c.add (pf);
}
```

勾 Logging and going to another frame

- নতুন Frame create করলে এবং রেজিস্টার করা হবে।
  - নতুন Frame retirement করুণায়ি সার্ভিস নিবে।
  - নতুন class এর মধ্যে ওজেট তৈরি করা হবে।
  - রেজিস্টার করা পর্যন্ত Frame অব্যাহত রাখা হবে।  
Object.setVisible(true);      দ্রুতেরে Frame delete  
করে আবে দিবে।

এখন কৃতির দিকে আসে একটি প্রক্রিয়া নাম ডিসপেজ (disposal)।

```
Test() {
 component();
}
public void com()
```

```
ActionListener() {
 if (Login button true)
 {
 newObject.setVisible(true);
 dispose();
 }
}
```

কোনো new Frame close  
বাধা করে old FR MR  
করেছে [DISPOSE\_ON\_CLOSE]

use ফুটবল

```
import java.awt.TextArea
```

## JTextArea

- Multi-line ग्रंथ लिखने का लिए
- अलग paragraph type
- JTextArea obj create करें तो Then  
बाहर स्क्रीन से नहीं आएगा → System

```
Test() {
```

```
 component();
```

```
}
```

```
public void component() {
```

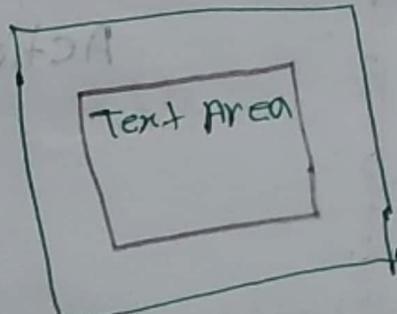
```
JTextArea A = new JTextArea();
```

```
A.setBounds();
```

```
A.setFont();
```

```
cn.add(A);
```

```
}
```



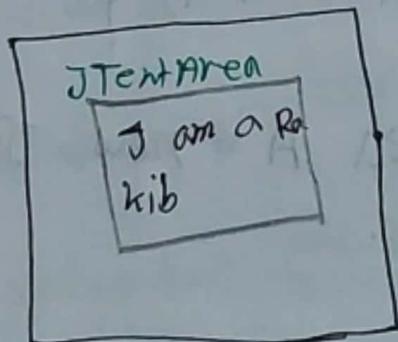
## Line Wrapping - JTextArea

- setLineWrap(true);
- setWrapStyleWord(true);

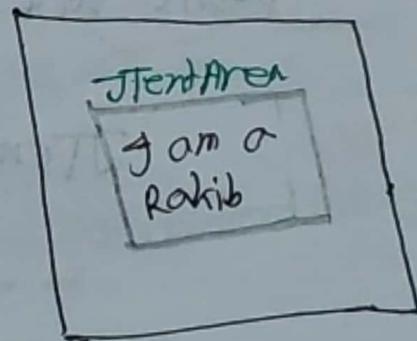
→ Se একান্তর ব্যাপ্তি JTextArea টেক্সট অফিস  
অন্তর আছেন একান্তর call করে মডেল অন্তর্ভুক্ত

Enter ২৫-৩৫

setLineWrap()



setWrapStyleWord()



JTextArea দ্বাৰা object. method টাৰ ২০১  
যোগ কৰত

```
import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.GridLayout;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
```

## 4) JScrollPane

- Object create করুন
- JScrollPane এর মধ্যে JTextArea এবং বাস্তব
- JTextArea কে কোথা দেখিয়ে থাকে scroll ফলে উন্নত  
fact, আব-বাব, দেখা সহজ।

```
Test () {
```

```
 component ()
```

```
}
```

```
public void component () {
```

```
 JTextArea A = new JTextArea ();
```

```
JScrollPane n = new JScrollPane (A);
```

```
 n. add (n);
```

```
}
```

```
[n = new JScrollPane (JScrollPane. vertical - scrollBarAlways,
 JScrollPane. horizontal - scrollBarAlways)]
```

```
import javax.swing.JPanel;
```

## ㈣ JPanel

- এমন object create করো কেন ?
- Java container ও তার ফলাফল কেন ?
- panel ও আবিন করে সব ব্যাখ্যা করো
- সব component add করল পায়ে panel কে

```
Test() {
```

```
 component();
```

```
}
```

```
public void component() {
```

```
 Container c = nothing.getContentPane();
```

```
 JPanel p1 = new JPanel
```

```
 c.add(p1);
```

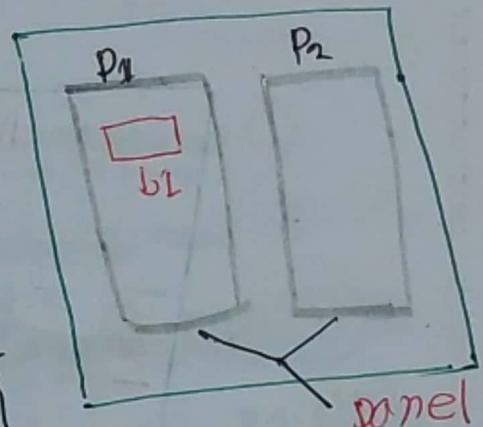
```
 JPanel p2 = new JPanel
```

```
 c.add(p2)
```

```
 JButton b1 = new JButton()
```

```
 p1.add(b1);
```

```
}
```



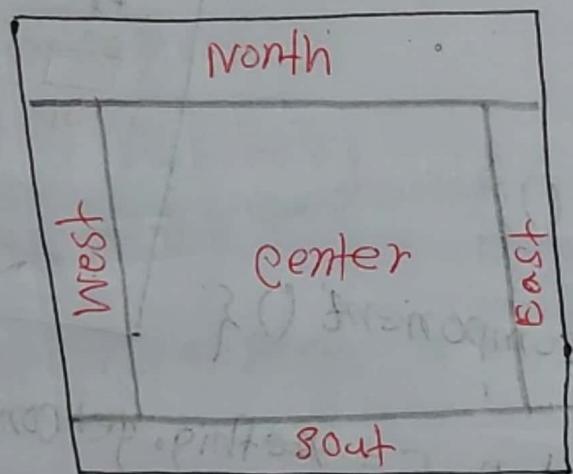
## important java.swing.BorderLayout;

### Layout Manager

→ Layout এর অনেক class আছে

→ Layout ব্যবহার করলে Bounds ব্যবহার করতে হবে

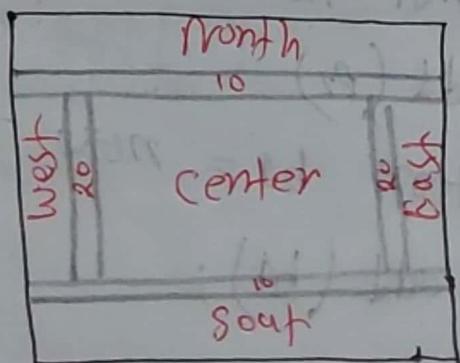
### BorderLayout



setHgap(10)

setVgap(20)

Border Button দ্বা-র মধ্যে gap মোট 30



```
test() { BorderLayout bl = new BorderLayout(1, 1);
 component(); } Layout w.r.t gap
```

```
public void component() { container c = this.getContentPane();
 BorderLayout bl = new BorderLayout();
```

```
c.setLayout(bl);
```

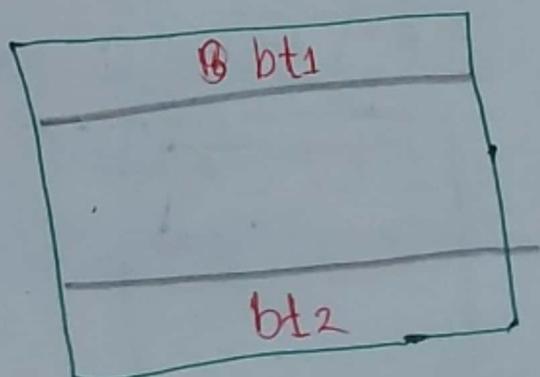
```
JButton bt = new JButton();
// adding method
```

```
c.add(bt, BorderLayout.NORTH);
```

```
c.add(bt, BorderLayout.SOUTH);
```

→ नाही एक add घेणे ज्या object

→ दोनों add घेणे ज्यां Location



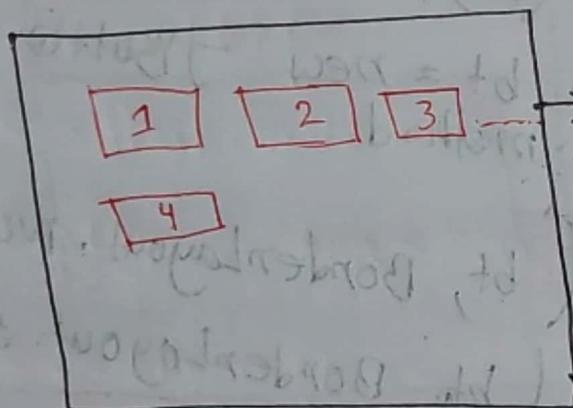
```
import javax.swing.FlowLayout;
```

## ■ FlowLayout

→ Word ଏକ ରୀତ ଯାଇଁ Horizontally add ହେଲେ

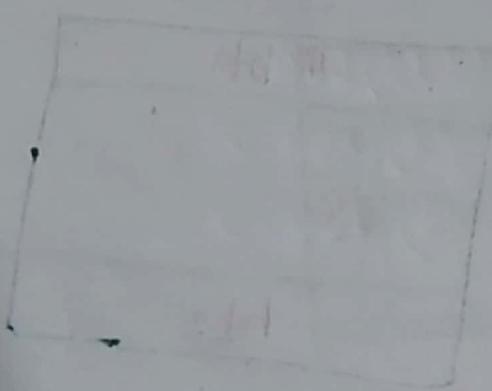
ଅଛି ଏହାମ ଲାଇଟ୍ କାମରୁ ନାହିଁ ତାହାର  
ବିଜ୍ଞାପନ ଲାଇଟ୍ ଥାଏ)

→ JPanel ଏକ Layout by default FlowLayout



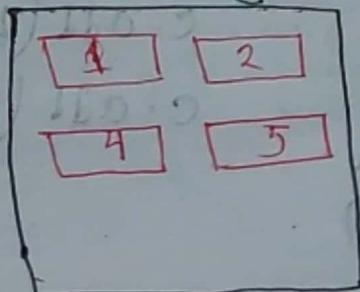
→ Object create କରିଲୁ ଥିଲୁ ।

→



```
Test () {
 component ();
}

public void component () {
 Container c = this.getContentPane();
 FlowLayout FL = new FlowLayout()
 c.setLayout (FL);
 JButton bt1 = new JButton ("1");
 JButton bt2 = new JButton ("2");
 c.add (bt1);
 c.add (bt2);
 or
 c.add (new JButton ("4"));
 c.add (new JButton ("5"));
}
```



```
import javax.swing.BoxLayout;
```

## BoxLayout

→ Button ফলো করে Horizontally or vertically

add করে।

→ একটি ব্যবহার করা হচ্ছে

Test

component () {

c = this.getContentPane();

BoxLayout bon = new BoxLayout(c, BoxLayout.Y\_AXIS);

c.setLayout(bon);

bt1 = new JButton("1");

bt2 = new JButton("2");

bt3 = new JButton("3");

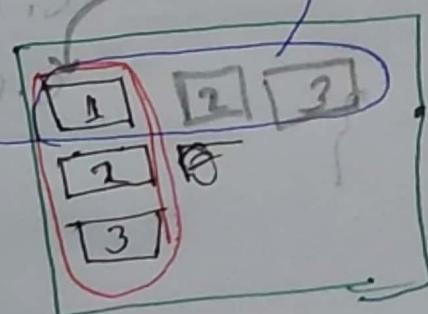
c.add(bt1);

c.add(bt2);

c.add(bt3);

}

BoxLayout.X\_AXIS



## 4) Grid Layout

→ Row एवं column का तारीफ करता है।

→ GridLayout एवं object से बहुत ज़्यादा लिंग।

```
Test() {
```

```
 component();
```

```
}
```

```
public void component() {
```

```
 c = this.getContentPane();
```

```
 GridLayout L = new GridLayout(2, 3);
```

~~C.addLayout~~

```
c.setLayout(L);
```

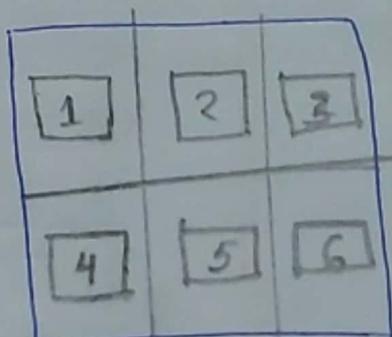
```
bt1 = new JButton("1")
```

```
.....
```

```
bt6 = new JButton("6")
```

```
c.add(bt1);
```

```
c.add(bt6);
```







# United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam, Trimester: Fall 2023

Course Code: CSE 1115, Course Title: Object Oriented Programming

Total Marks: 40, Duration: 2 Hours

*Any examinee found adopting unfair means will be expelled from the trimester/program as per UIU disciplinary rules.*

Q1:

(a) Consider the interface *Transaction*

```
public interface Transaction {
 public void give();
 public void receive();
}
```

[4+4]

The *Payment* class implements interface *Transaction*. Code for the *Payment* class is provided as follows:

```
public abstract class Payment implements Transaction{
 double amount;
 String currency ="BDT";
 @Override
 public void give(){
 //Write code here
 System.out.println(currency+" "+amount+" is paid");
 }
 @Override
 public void receive() {
 System.out.println("[Nothing to receive]");
 }
 public abstract double getCharge();
 public double getTax(){
 //Write code here
 }
}
```

Create three classes *CashPay*, *CreditCardPay* and *CheckPay* that are subclasses of *Payment*. Complete the code for *Payment* class. Write all the necessary codes according to the following description:

- I. There is no charge added for cash payment.
- II. 2.5% charge is added to the amount for credit card payment.
- III. 10 taka flat charge is added for payment with check.
- IV. 15.6% VAT is added for all types of payment
- V. *give()* method prints the amount to be given including charge and VAT.

(b) Consider the following code.

```
public interface Shape {
 double getArea();
}
class Rectangle implements Shape {
 private double length, width;

 public Rectangle(double length, double width) {
 this.length = length;
 this.width = width;
 }
 @Override
 public double getArea() {
 return length * width;
 }
}

class Square extends Rectangle{
 //write your code here
```

```

 }
class ShapeTest{
 //write code for draw method here
 public static void main(String[] args) {
 Shape r = new Rectangle(5,6);
 Shape s = new Square(3);
 draw(r);
 draw(s);
 }
}

```

M. Complete the code for the Square class.

~~T~~ D Implement the method *draw* in ShapeTest class so the following output is produced.

|                        |
|------------------------|
| drawing over 30.0 area |
| drawing over 9.0 area  |

Q2:

(a) Suppose that you are given a text file named "input.txt" with random texts. Now you have to check how many consonants are in the file and write it to another text file named "output.txt". For example, if the input file contains the following text - "Don't be upset", you should write "7" in the output file. Both the files should be in the "src" directory. Now, write a program in Java to perform this task. [6+6]



- What happens if an exception is not caught or handled in Java?
- Write the output of the following code.

```

import java.util.*;
import java.lang.*;
import java.io.*;

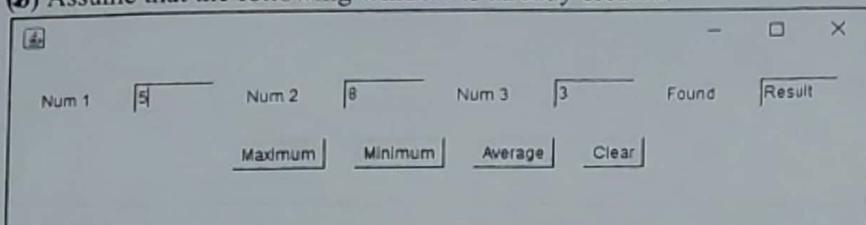
public class Main{
 public static void main (String[] args) throws java.lang.Exception{
 int arr[] = new int[4];
 int x = 10, y = 1;
 try{
 try{
 arr[4] = x / (y - 1);
 System.out.println("a");
 }
 catch(ArithmaticException e){
 System.out.println("b");
 arr[4] = x / (y - 1);
 }
 catch(IndexOutOfBoundsException e){
 System.out.println("c");
 arr[4] = x / (y - 1);
 }
 finally{
 System.out.println("d");
 }
 }
 catch(Exception e){
 System.out.println("e");
 }
 finally{
 System.out.println("f");
 }
 }
}

```

(a) Suppose that you are required to write a Java program for implementing a GUI using any layout. GUI initially contains a single button named *ADD*. *ADD* button dynamically adds a new button to the frame when it is clicked. Each new button should have a unique label too (such as *b1*, *b2*, *b3* ...). Note that the buttons or labels should not overlap. Now do the following task to demonstrate your skill:

Write only the codes to create the user interface and an event-handling method for *ADD* button.

(b) Assume that the following window is already created.



The names for text field variable are *n1*, *n2*, *n3*, and *result*. For buttons, the names are *max*, *min*, *avg*, and *clr*.

Now write the proper event handling method to do the following task,

- A. After clicking the maximum button, the *result* field should show the maximum of *n1*, *n2*, and *n3*.
- B. The minimum button will set the minimum value of *n1*, *n2*, and *n3* in the *result* field.
- C. Average button will find the average of all these inputs.
- D. Clear button will clear all the text fields.

Q4:

Consider the following Address class:

```
public class Address {
 String building_number, area, city;
 int zip_code;

 Address(String building_number, String area, String city, int zip_code){
 this.building_number = building_number;
 this.area = area;
 this.city = city;
 this.zip_code = zip_code;
 }

 int getZip_code(){
 return zip_code;
 }
}
```

Based on the above class, complete the following tasks:

```
class Test{
 public static void main(String[] args){
 //Task 1: Create an empty ArrayList of Address type

 /*
 Task 2: Add the following objects in the ArrayList
 "19/A", "Dhanmondi", "Dhaka", 1209
 "2/A", "Tejgaon", "Dhaka", 1215
 "65", "Nirala", "Khulna", 9100
 */

 /*
 Task 3: Add the below object at index 1 of the ArrayList
 "215", "Aamtola", "Barishal", 8200
 */

 /*
 Task 4: Set the object at index 2 to
 }
```

```
"36", "Gulshan", "Dhaka", 1212
*/
/*
Sort the arraylist in, ascending order of zip codes using a
comparator for comparing objects of Address class [You can also
define the Comparator as a separate class if you want]
*/
/*
Task 5: Delete the object at index 2
*/
}
}
```

- Q5: Asfaq, a first-year programming student, learned how to find the highest number from an array using the Java program. Excited with his newfound knowledge, he started to wonder if he could concurrently do the same for multiple arrays. One of his seniors told him to use Thread which facilitates to run multiple tasks concurrently. Based on the idea, in order to help Asfaq, your task is to
- write a Java program that can find the highest numbers from 4 integer arrays concurrently and next, compute the maximum of these highest numbers and print the maximum number. Consider the arrays as {3, 1, -5, 10}, {-2, 6, 7, 8, 0}, {12, -6, 4, 2, 1}, {10, 5, -9, 18, 7}.

[6]