



CST EDU CARE .BTEB
ONLINE CLASSROOM
ESTD : 2025

APPLICATION DEVELOPMENT USING JAVA

FINAL SUGGESTION



++
++
++

Prepared by:
MD. ABIR HOSSEN

+8801930921578

<https://cse-edu-care-bteb.netlify.app/>



Application Development Using Java

কেন্দ্রীয়

1. আয়তক্ষেত্রের ক্ষেত্রফল নির্ণয় - Java program নিম্ন।

```
import java.util.Scanner;  
public class Rectangle {  
    public static void main (String [] args) {  
        double Area, length, width;  
        Scanner input = new Scanner (System.in);  
        System.out.print ("Enter the length value=");  
        length = input.nextDouble();  
        System.out.print ("Enter the width value=");  
        width = input.nextDouble();  
  
        Area = length * width;  
        System.out.println ("The Rectangle Area=" + Area);  
    }  
}
```

2. বৃত্তের ক্ষেত্রফল নির্ণয় - Java program নিম্ন।

```
import java.util.Scanner;  
public class Circle Area {  
    public static void main (String [] args) {  
        double Area, Radius;  
        Scanner input = new Scanner (System.in);  
        System.out.print ("Enter the Radius=");  
        Radius = input.nextDouble();
```

Area = Math.PI * Radius * Radius;

System.out.println("The Circle Area = " + Area);

}

}

3. Constructor Method ব্যবহার করে Java Program

মিঃ -

```
public class Student {
```

```
    int Roll;
```

```
    String Name;
```

```
    float CGPA;
```

```
    public Student() {
```

```
        Roll = 732309;
```

```
        Name = "Abir Hossen";
```

```
        float CGPA = 3.95;
```

```
        System.out.println("The name = " + name);
```

```
        System.out.println("The Roll = " + Roll);
```

```
        System.out.println("The CGPA = " + CGPA);
```

```
    }
```

```

public class main {
    public static void main (String [] args) {
        Student S = new Student ();
    }
}

```

4. Copy Constructor बनाकर Java Program

```

public class Student {
    String Name ;
    int Roll ;

    public Student (String N , int R) {
        Name = N ;
        Roll = R ;
    }

    public Student S (Student) {
        Name = This . Name ;
        Roll = This . Roll ;
    }

    public void display () {
        System.out.println ("The Name =" + Name);
        System.out.println ("The Roll =" + Roll);
    }
}

```

```

public class copyConstructor {
    public static void main (String [] args) {
        Student s1 = new Student ();
        Student s2 = new Student ();
        s1.display ();
        s2.display ();
    }
}

```

5. Method Overloading

```

public class Abir {
    public int cal (int i, int j) {
        int k;
        k = i + j;
        System.out.println ("Addition value = " + k);
    }
    public float cal (float x, float k) {
        float y;
        y = x + k;
        System.out.println ("Addition value = " + y);
    }
}

```

```
public bool Cal () {  
    int x ;  
    if(x=0) {  
        System.out.println ("True");  
    } else {  
        System.out.println ("false");  
    }  
}
```

```
public class example {
```

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

```
        Abir A = new Abir ();
```

```
        A.Cal (10, 20);
```

```
        A.Cal (3.75, 3.95);
```

```
        A.Cal (1);
```

```
    }  
}
```

6. Interface বাস্তবায়নের Java Program লিখ।

```
Interface Animal {  
    public void MakeSound () {  
    }  
}  
  
public class Dog implements Animal {  
    public void MakeSound () {  
        System.out.println ("Meow ! Meow !");  
    }  
}  
  
public class Cat implements Animal {  
    public void MakeSound () {  
        System.out.println ("Woof ! woof !");  
    }  
}  
  
public class Interface Ex {  
    public static void Main (String [] args) {  
        Animal my Dog = new Dog ();  
        Animal my Cat = new Cat ();  
    }  
}
```

7. Super class এবং subclass ব্যবহার করে Java program

লিখ।

```
class Animal {
```

```
    void sound() {
```

```
        System.out.println("Animal make a sound");
```

```
    }
```

```
}
```

```
class Dog extends Animal {
```

```
    void sound() {
```

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

```
    }
```

```
}
```

```
public class Main {
```

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

```
        Dog d = new Dog();
```

```
        d.sound();
```

```
    }
```

```
}
```

8. উদাহরণ অর্থাৎ String Method গুলো বর্ণনা করা।

Java-তে অনেকগুলো String Method রয়েছে। যার
মধ্যে উল্লেখযোগ্য হারটি Method নিচে বর্ণনা করা হলো:

(i) length(): এ Method এর মাধ্যমে String এর দৈর্ঘ্য
জানা যায়।

উদাহরণ:

```
String str = "hello";  
int length = str.length();  
System.out.println(length);
```

Output: 5

(ii) charAt(int index): এ Method টি String এর নির্দিষ্ট
অবস্থান থেকে ক্যারেক্টার প্রদর্শন করে।

উদাহরণ:

```
String str = "Afron";  
char ch = str.charAt(2);  
System.out.println(ch);
```

Output: r

(iii) equals(Object obj): এ Method টি দুইটি String
সমান কিনা তা পরীক্ষা করার জন্য ব্যবহৃত হয়।

উদাহরণ:

```
String S1 = "Abir";  
String S2 = "Tuya";  
boolean eq = S1.equals(S2);  
System.out.println(eq);
```

Output: True

(iv) CompareTo (String another String): এই method টি দুইটি String এর মধ্যে তুলনা করে কোন পক্ষের String টি-র String থেকে ছোট, বড় নাকি সমান তা প্রকাশ করে।

উদাহরণ:

```
String S1 = "Abir Hossen";  
String S2 = "Tuya Islam";  
int Comp = S1.compareTo(S2);  
System.out.println(comp);
```

Output: 1 Big;

9. উদাহরণ সহ Java বিক্রমজন হ্যান্ডলিং প্রোটোকল বর্ণনা কর।

Exception Handling হলো কোন একটি পত্রা বা program কে স্থিতিশীল ও দক্ষ করে তোলে। Exception Handling এর মাধ্যমে programmer বা Error কমান্ড ও তা পরিচালনা করতে পারে। নিচে Exception Handling এর ধাপ-সমূহ বর্ণনা করা হলো:

Step 1: সম্ভাব্য Exception কমান্ড করতে হবে।

প্রথমে programmer দেব তাদের program এ সম্ভাব্য Exception সূচনা কমান্ড করতে হবে। Exception সর্বাধিকতম তখন ঘটে যখন কোন অস্বাভাবিক বা অপ্রত্যাশিত ঘটনা ঘটে যায়, যেমন কোন সংখ্যাকে 0 দ্বারা ভাগ করার চেষ্টা।

Step 2: Exception পরিচালনার জন্য পরিকল্পনা করতে হবে।

কিছোর Exception কমান্ড হয়ে গেলে programmer দেব কিভাবে Exception পরিচালনা করতে হবে তা পরিকল্পনা করতে হবে। Java programming এ Exception পরিচালনার দুইটি পত্রা রয়েছে।

- (i) try-catch ব্লক
- (ii) Throws বিধি।

Step 3: try-catch ব্লক ব্যবহার করতে হবে।

try-catch ব্লক হলো একটি কমান্ড যা একটি Exception সনাক্ত হলে তা কিভাবে পরিচালনা করতে হবে তা নির্দেশ দেয়। try-catch ব্লক দুইটি অংশ নিয়ে গঠিত।

- (i) try অংশ: ব্লক অংশে কোডটিকে অনর্দ্র করে যা Exception সৃষ্টি করতে পারে।
- (ii) catch অংশ: ব্লক অংশে কোডটিকে অনর্দ্র করে যা Exception পরিচালনা করে।

Step 4: throws বিধি ব্যবহার করতে হবে।

throws বিধি হলো একটি command যা Exception সৃষ্টি করে একাধিক অংশে অন্য পদ্ধতি বা ক্লাসে পাঠানো করে।

Step 5: programটি পরীক্ষা করতে হবে।

বিভিন্ন programmer বা Exception Handling Code লিখে ফোলো তা পরীক্ষা করতে হবে যে এটি কিভাবে ডায়েকাজ করছে কিনা।

10. Java Program এর সাথে Database Connection এর ধাপ সমূহ লেখ।

Register Driver

1

Get Connection

2

Create Statement

3

Execute Query

4

Close Connection

5

উপরে Java program এর সাথে Database Connection
এর ধাপসমূহ চিত্রাকারে দেখানো হয়েছে। নিচে এ ধাপ সমূহের
বর্ণনা দেওয়া হলো।

(1) Register Driver : প্রথমে যা Database এর সাথে আনুষঙ্গিক
করাতে হবে তার জন্য JDBC Driver ইন্স্টল করতে হবে।
Driver এর Developer এর Website থেকে Driver
Download করা যাবে। Driver Download করা হয়ে গেলে
তা ইন্স্টল ও রুটিংস্টেবল করতে হবে।

(II) Get Connection: JDBC API'র মাধ্যমে Database এর সাথে আয়োগ করার জন্য একটি Connection Object তৈরি করতে হবে। Connection Object Database এর সাথে একটি আয়োগ প্রতিনির্বিষ্ট করবে। Connection Object তৈরির জন্য DriverManager Class এর getConnection() Method ব্যবহার করতে হবে।

(III) Create Statement: Connection Object ব্যবহার করে SQL Statement তৈরি ও কার্যকর করা যেতে পারে। SQL Statement হলো একটি কমান্ড যা Database কে কী করতে হবে তা নির্দেশ দেয়। Statement Object তৈরি করতে Connection Object এর createStatement() Method ব্যবহার করতে হবে।

(IV) Execute Query: SQL Statement কার্যকর করতে Statement Object এর execute() Method ব্যবহার করতে হবে। Execute() Method চি একটি Boolean মান ফেরত দেয় যা নির্দেশ করে যে Statement চি সফলভাবে কার্যকর হয়েছিল কিনা।

(V) Close Connection: JDBC API'র মাধ্যমে Database এর সাথে আয়োগ শেষ করার জন্য আয়োগ বন্ধ করতে হবে। আয়োগ বন্ধ করতে Connection Object এর close() Method ব্যবহার করতে হবে।

11. উদাহরণস্বরূপ Java Swing Component গ্রহণ করে আলোচনা কর।

Java ত ইন্টারফেস/ইন্টারঅ্যাকটিভ ইন্টারফেস তৈরি জন্য যে একমাত্র ডিউয়াল Control Component ব্যবহার করা হয় তাকে Swing Component বলে। Java ত এক্ষেত্রে Swing Component আছে যেগুলো ব্যবহার করে ইন্টারফেস/ইন্টারঅ্যাকটিভ Java Application তৈরি করা হয়। নিচে তার বর্ণনা দেওয়া হলো:

JFrame Component: JFrame Component এক মার্জিত কোন GUI Application এর ক্ষেত্রে উইন্ডো তৈরি করা যায়। যাতে একটি-টাইটেল বার, মিনিমাইজ বার্টেন ও ম্যাক্স বার্টেন ডিফল্টভাবে থাকে। JFrame Component এর সিনটাক্স হলো:

```
JFrame frame = new JFrame("FrameExample");
```

JPanel Component: এ Component সাধারণত একসাথে সমন্বিত Component গুলোকে গ্রুপ আকারে বিস্তারিত করার জন্য ব্যবহৃত হয়। JPanel Component এর সিনটাক্স হলো:

```
JPanel panel = new JPanel();
```

```
panel.setBounds(40, 80, 200, 200);
```

```
panel.setBackground(Color.GRAY);
```

JButton Component : এ Component ব্যবহারে কোন Application এ ক্লিক প্রোগ্রাম Button তৈরি করা হয়। এর syntax হলো:

```
JButton b1 = new JButton ("click");  
b1. setBounds (50, 100, 80, 30);  
b1. setBackground (color, blue);
```

JLabel Component : এটি User Interface এ কোন Read Only text বা Image প্রদর্শনের জন্য ব্যবহৃত হয়। JLabel Component এর syntax হলো:

```
JLabel text = new JLabel ("Name");  
JLabel img = new JLabel ("car icon");
```

JTextField Component : আবাসিত user হতে কোন নন-ব্যবস্যাভেঁ Text input নিতে এই Component ব্যবহার করা হয়। এর সিনট্যাক্স হলো:

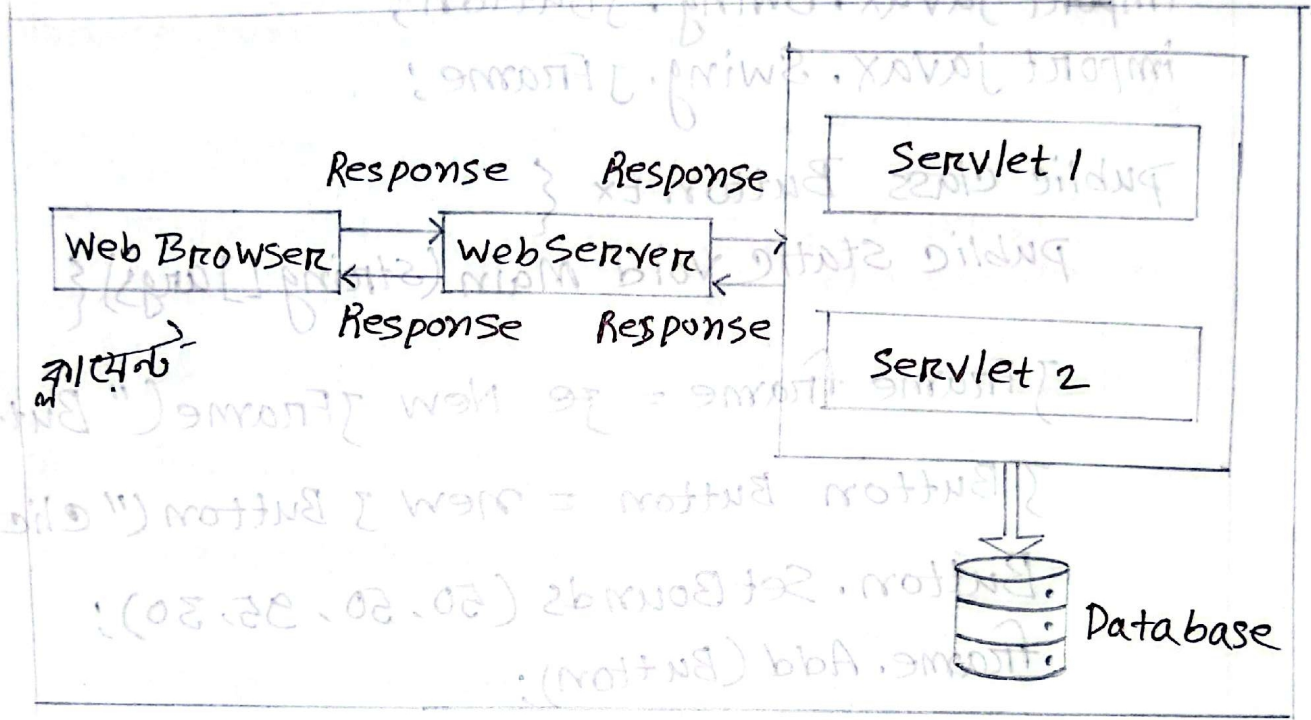
```
JTextField t1 = new JTextField ("welcome");
```

12. Java Swing Component ବ୍ୟବହାର କରା Button
ଉଦାହରଣ - Java program ନିମ୍ନ ।

```
import javax.swing.JButton;  
import javax.swing.JFrame;
```

```
public class ButtonEx {  
    public static void main (String [] args) {  
        JFrame frame = new JFrame ("ButtonEx");  
        JButton button = new JButton ("click");  
        button.setBounds (50, 50, 95, 30);  
        frame.add (button);  
  
        frame.setSize (300, 200);  
        frame.setLayout (null);  
        frame.setVisible (true);  
        frame.setDefaultCloseOperation (JFrame.EXIT-  
            ON-CLOSE);  
    }  
}
```

13. লুক ডায়াগ্রাম অর্থাৎ সার্ভলেট আর্কিটেকচার কার্যনীতি-
 ব্যাখ্যা কর।



উপরের চিত্রে Component অর্থাৎ Servlet Architecture
 এর লুক লুক ডায়াগ্রাম দেখানো হয়েছে। Servlet এর Component

সমূহ হলো:

- (I) ক্লায়েন্ট
- (II) ওয়েব সার্ভার
- (III) সার্ভলেট কন্ট্রোলার
- (IV) সার্ভলেট

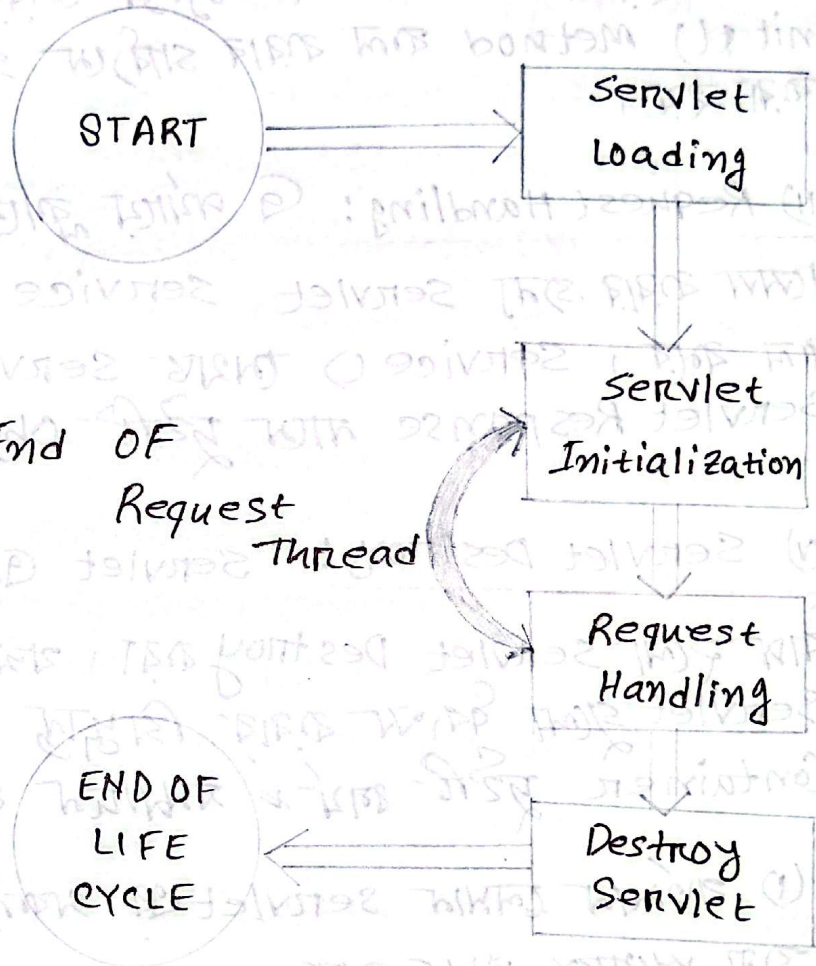
(III) Servlet সূত্রের Java Servlet সূত্রের সাথে
যোগাযোগ করে।

(IV) Java Servlet Database থেকে Data Access
ও Manipulate করে।

(V) Java Servlet Servlet Container এর নিষ্কাশন
বৈশিষ্ট্য - পাঠ্য।

(VI) Java Servlet Container বৈশিষ্ট্যটি গ্রহণ
করে তা ক্রমিকভাবে বসে দেয়।

14. java Servlet এর জীবনচক্র বর্ণনা কর।



উপরের চিত্রের মাধ্যমে java servlet এর জীবনচক্র দেখানো হয়েছে। java Servlet এর LIFE CYCLE চারটি ধাপে সম্বন্ধ রয়েছে। নিচে ধাপ সমূহ বর্ণনা করা হলো:

(1) Servlet Loading : Servlet এর life cycle শুরু হয় Servlet Loading এর মাধ্যমে। Servlet Container দ্বারা Servlet Load করা হয়।

(ii) Servlet Initialization: Servlet এর life cycle এর second step is Servlet Initialization করা হয়। Servlet ইনস্ট্যান্স তৈরি পর Web Container `init()` Method কাল করার মাধ্যমে Servlet Initialization করা হয়।

(iii) Request Handling: এ পর্যায় ক্লায়েন্টের Request প্রাপ্ত করার জন্য Servlet Service Method `Service()` কাল করে। `Service()` মেথডে Servlet Request ও Servlet Response নাম দুইটি Object তৈরি করে।

(iv) Servlet Destroy: Servlet এর জীবন চক্রের অবশেষে সার্ভিসে Servlet Destroy করা। যখন Servlet Container Servlet সুলো করা করার সিদ্ধান্ত নেয় তখন Servlet Container দুইটি কার্যকর সম্বাদন করে।

(i) বর্তমান চলমান Servlet এর সকল কাজ বন্ধ করার জন্য সুযোগ প্রদান করে।

(ii) সকল কার্যক্রমের পর `Destroy()` মেথডকে কাল করে ও Servlet কাল করে দেয়।