

# জাভা দিয়ে লজিক বিল্ডিং

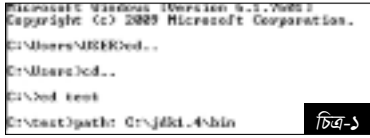
মো: আবদুল কাদের .....

জনপ্রিয় প্রোগ্রামিং ল্যান্ডমার্ক জাভা। জাভার সিকিউরিটি, হাই পারফরম্যান্স ও কোড ফাইলের আকার খুব ছোট হওয়ায় এর ব্যবহার ব্যাপক। এছাড়া যেকোনো অপারেটিং সিস্টেমে রান করাই জাভার প্রধান বৈশিষ্ট্য। এ কারণেই ওয়েবসাইট বিল্ডিং, মোবাইল অ্যাপ্লিকেশন, মোবাইল গেম তৈরি, চ্যাটিং সফটওয়্যারসহ প্লাটফর্ম ইন্ডিপেনডেন্ট কাজে জাভার প্রয়োগ বেশি। যেকোনো প্রোগ্রাম বা অ্যাপ্লিকেশন ডেভেলপ করতে লজিক গুরুত্বপূর্ণ ভূমিকা পালন করে। এ পর্বে জাভা দিয়ে কিছু গুরুত্বপূর্ণ লজিক বাড়াবোর কৌশল দেখানো হয়েছে।

## কীবোর্ড থেকে ইনপুট নেয়া

জাভা প্রোগ্রামে কীবোর্ড থেকে ইনপুট নেয়া যায় তিনটি উপায়ে : ০১. কোনো প্রোগ্রাম রান করার সময়, ০২. প্রোগ্রাম রান করার পরে বা চলা অবস্থায় ইউজারের কাছ থেকে ইনপুট নেয়া ও ০৩. উইন্ডোভিত্তিক কোনো অ্যাপ্লিকেশনে যেমন টেক্সটবক্স, টেক্সট এরিয়া ইত্যাদিতে ইনপুট দেয়া। এ পর্বে প্রোগ্রাম রান করার সময় ইনপুট দেয়ার দুটি প্রোগ্রাম ও রানিং অবস্থায় ইনপুট নেয়ার একটি প্রোগ্রাম দেয়া হলো।

প্রথমেই আমরা ইউজারের কাছ থেকে নেয়া কয়েকটি সংখ্যার মধ্যে বড় সংখ্যাটি খুঁজে বের করার প্রোগ্রাম দেখব। এই প্রোগ্রাম রান করার সময় ইউজার কয়েকটি নাম্বার দিলে সবচেয়ে বড় নাম্বারটি দেখাবে। নিচের প্রোগ্রামটি নোটপ্যাডে টাইপ করে FindMax.java নামে সেভ করে চিত্র-২-এর মতো করে রান করতে হবে। তবে এর আগে ইনস্টল করা জাভা সফটওয়্যার (jdk1.4)-এর ড্রাইভ দেখিয়ে দেয়ার জন্য চিত্র-১ অনুসরণ



চিত্র-১

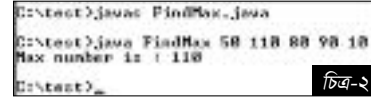
করুন। এ লেখায় ফাইলগুলো C ড্রাইভের test ফোল্ডারে সেভ করা হয়েছে।

```
class FindMax
{
public static void main(String args[] ) //1
{
int max=0;
int i[ ] = new int[args.length]; //2
for (int k=0; k<args.length; k++)
{
i[k]=Integer.parseInt(args[k]); //3
}
max=i[0]; //4
for(int j=1; j<i.length; j++)
{
if(max<i[j]) //5
max=i[j];
}
System.out.println("Max number is : "+max); //6
}
}
```

## কোড বিশ্লেষণ

প্রোগ্রামটির শুরুতে ১নং চিহ্নিত লাইনে main মেথডের আর্গুমেন্ট হিসেবে String টাইপের অ্যারে args ডিক্লেয়ার করা হয়েছে। রান টাইমে ইউজার ইনপুট গ্রহণ করার জন্য এটি ব্যবহার করা হয় এবং ইনপুটগুলো স্ট্রিং বা টেক্সট হিসেবে নেয়া হয়। এমনকি নাম্বার দিলেও তা টেক্সট হিসেবে গণ্য হয়। প্রথম ইনপুটটি args অ্যারের ০ পজিশনে (args[0]), পরেরটি ১ পজিশনে (args[1]) এভাবে ক্রমান্বয়ে ইনপুটগুলোকে সাজানো হয়। প্রোগ্রামে বড় সংখ্যাটি রাখতে বা প্রিন্ট করার জন্য max ভেরিয়েবল ও args অ্যারের ইনপুটগুলো নাম্বারে পরিবর্তন করে রাখার জন্য ২নং লাইনে

ইন্টিজার টাইপের অ্যারে i নেয়া হয়েছে। অ্যারেতে কতগুলো ভেরিয়েবল তৈরি হবে, তা নির্দিষ্ট করে না দিয়ে ইউজার যতগুলো সংখ্যা দেবে সে সংখ্যক ভেরিয়েবল তৈরির জন্য args.length ব্যবহার করা হয়েছে। এরপর for লুপ ব্যবহার করে Integer.parseInt-এর মাধ্যমে ৩নং লাইনে একটি করে ইনপুট নাম্বারে পরিবর্তন করে i অ্যারেতে রাখা হচ্ছে। আমরা প্রথমত i অ্যারের i[0] পজিশনের নাম্বারটিকে বড় ধরে নিচ্ছি এবং তা ৪নং লাইনে max ভেরিয়েবলে রেখে আরেকটি for লুপের সাহায্যে পরের নাম্বারগুলোর সাথে তুলনা করা হবে। ৫নং লাইনে if কন্ডিশন তৈরি করা হয়েছে। এখানে max ভেরিয়েবলের মান i[1] পজিশনের চেয়ে বেশি হলে নাম্বারটি max-এ রাখবে। এভাবে max-এর সাথে ক্রমান্বয়ে i[2], i[3] ও i[4] নাম্বারগুলো তুলনা করে বড় সংখ্যাটি বের করা হয়, যা ৬নং লাইনে প্রিন্ট করা হচ্ছে।



চিত্র-২

চিত্র-২-এর প্রথম লাইনে প্রোগ্রামটিকে কম্পাইল করে দ্বিতীয় লাইনে প্রোগ্রামটি রান করার সময় আমরা পাঁচটি নাম্বার ব্যবহার করেছি। এর মধ্যে সবচেয়ে বড় সংখ্যাটি ছিল 110, সেটি পরের লাইনে প্রিন্ট করেছে। ইউজার ইচ্ছে করলে পাঁচটির বেশি নাম্বারও ব্যবহার করতে পারবেন।

## দ্বিতীয় প্রোগ্রাম

এই প্রোগ্রাম রান করার সময় ইউজার যে নাম্বার বা টাকার অঙ্ক দেবেন তাকে ভাগ্যতে কয়টি নোট (বড় সংখ্যার নোট হতে ছোট সংখ্যার) দরকার তা দেখাবে।

নিচের প্রোগ্রামটি নোটপ্যাডে টাইপ করে ConvrtInNote.java নামে সেভ করে চিত্র-৩-এর মতো করে রান করতে হবে।

```
public class ConvrtInNote
{
public static void main(String args[])
{
int t500=0, t100=0, t50=0, t20=0, t10=0, t5=0, t2=0, t1=0;
int a = Integer.parseInt(args[0]); //1
System.out.println(a + " taka is converting in note");
System.out.println("-----");
t500=a/500; //2
if (t500 !=0)
System.out.println("500 Taka note need : " + t500 + " piece");
t100=(a-(t500*500))/100; //3
if (t100 !=0)
System.out.println("100 Taka note need : " + t100 + " piece");
t50=(a-(t500*500 + t100*100))/50; //4
if (t50 !=0)
System.out.println("50 Taka note need : " + t50 + " piece");
t20=(a-(t500*500 + t100*100 + t50*50))/20; //5
if (t20 !=0)
System.out.println("20 Taka note need : " + t20 + " piece");
t10=(a-(t500*500 + t100*100 + t50*50 + t20*20))/10; //6
if (t10 !=0)
System.out.println("10 Taka note need : " + t10 + " piece");
t5=(a-(t500*500 + t100*100 + t50*50 + t20*20 + t10*10))/5; //7
if (t5 !=0)
System.out.println("5 Taka note need : " + t5 + " piece");
t2=(a-(t500*500 + t100*100 + t50*50 + t20*20 + t10*10 + t5*5))/2; //8
if (t2 !=0)
System.out.println("2 Taka note need : " + t2 + " piece");
t1=a-(t500*500 + t100*100 + t50*50 + t20*20 + t10*10 + t5*5 + t2*2); //9
if (t1 !=0)
System.out.println("1 Taka note need : " + t1 + " piece");
}
}
```

চিত্র-৩

## কোড বিশ্লেষণ

প্রোগ্রামটিতে আট ধরনের নোট রাখার জন্য ৮টি ইন্টিজার টাইপের ভেরিয়েবল নেয়া হয়েছে। ১নং চিহ্নিত লাইনে কীবোর্ড থেকে নেয়া ইনপুট কনভার্ট করে a নামের ভেরিয়েবলে রাখা হয়েছে। এরপর ২, ৩, ৪, ৫, ৬, ৭, ৮ ও ৯নং লাইনে লজিক সেট করে ক্রমান্বয়ে ৫০০, ১০০, ৫০, ২০, ১০, ৫, ২ ও ১ টাকার নোটে পরিবর্তন করা হয়েছে, যা পরবর্তী লাইনগুলোর মাধ্যমে প্রিন্ট করা হয়েছে। প্রোগ্রামটি দেখলে খুব সহজেই বোঝা যাবে।

## রান টাইমে

### ইন্টারেকটিভ ইনপুট

প্রোগ্রামটি নোটপ্যাডে

টাইপ করে

InteractiveCalc.java নামে

সেভ করে চিত্র-৪-এর মতো রান করাতে হবে।

```
import java.io.*;
class InteractiveCalc
{
public static void main(String args[])
{
int num1=0, num2=0;
char sign='';
try
{
InputStreamReader isr=new InputStreamReader(System.in);
BufferedReader br = new BufferedReader(isr);
num1 = Integer.parseInt(br.readLine());
num2 = Integer.parseInt(br.readLine());
sign=(char)System.in.read();

```



চিত্র-৪

```
}
catch(Exception e){}
switch(sign)
{
case '+':
System.out.println("The sum is = "+ (num1+num2));
break;
case '-':
System.out.println("The subtraction is = "+ (num1-num2));
break;
case '*':
System.out.println("The multiply is = "+ (num1*num2));
break;
case '/':
System.out.println("The division is = "+ (num1/num2));
break;
default: System.out.println("Correct your input");
}
}
```

## কোড বিশ্লেষণ

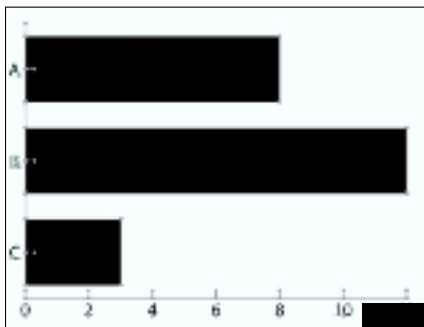
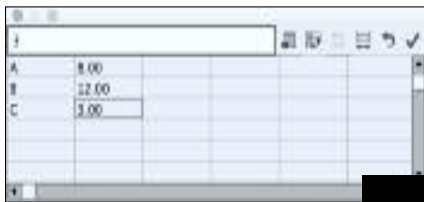
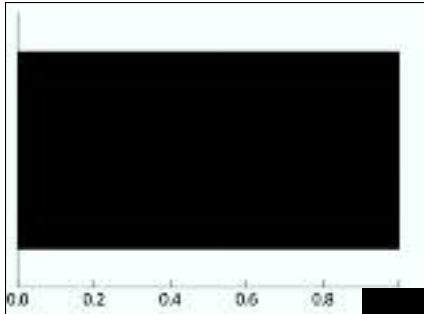
প্রোগ্রামটি রান করলে রানিং অবস্থায় ইউজারকে তিনটি ইনপুট দিতে হবে। প্রথম দুটি নাম্বার ও পরেরটি চিহ্ন (যোগ, বিয়োগ, গুণ বা ভাগ করার জন্য)। এখানে দুটি নাম্বার রাখার জন্য দুটি ইন্টিজার টাইপের ভেরিয়েবল num1 ও num2 এবং একটি চিহ্ন রাখার জন্য ক্যারেক্টার টাইপের ভেরিয়েবল sign নেয়া হয়েছে। তারপর switch case-এর মাধ্যমে ইউজারের দেয়া চিহ্ন অনুযায়ী যোগ, বিয়োগ, গুণ বা ভাগ হয়ে রেজাল্ট প্রিন্ট হবে।

পরে উইন্ডোভিত্তিক অ্যাপ্লিকেশন নিয়ে আলোচনা করা হবে **কল**

ফিডব্যাক : [balaith@gmail.com](mailto:balaith@gmail.com)

## ইলাস্ট্রেটরে চার্ট আঁকা

(৮৯ পৃষ্ঠার পর)



করতে হবে। তাহলে গ্রাফটি পরিবর্তন হয়ে চিত্র-৭-এর মতো হবে। এবার গ্রাফে রাইট ক্লিক করে কলাম অপশন সিলেক্ট করলে চিত্র-৮-এর মতো একটি উইন্ডো আসবে। এখানে বাম দিক থেকে রেড স্টার সিলেক্ট করলে চিত্র-৯-এর মতো একটি কাস্টম গ্রাফ পাওয়া যাবে। এবার গ্রাফটি ক্লিনাপ করার সময়। নিচের দিকে অর্থাৎ এক্স এক্সিসে যে স্কেল আছে সেটি রিমুভ করে দিলে সুন্দর একটি গ্রাফ পাওয়া যাবে। এজন্য সিলেকশন টুল দিয়ে গ্রাফটিকে সিলেক্ট করে প্রথমে অবজেক্ট → আনগ্রুপ সিলেক্ট করলে গ্রাফের প্রতিটি এলিমেন্ট আলাদা হয়ে যাবে।

এবার সাইড টুলবার থেকে ডিরেকশন সিলেকশন টুল দিয়ে অপ্রয়োজনীয় অংশগুলো সিলেক্ট করে ডিলিট করলে সেগুলো মুছে যাবে। এভাবে ইউজার নিজের মতো একটি কাস্টম পিক্টোগ্রাফ তৈরি করতে পারেন। ইউজার চাইলে একই গ্রাফে একাধিক আইকন ব্যবহার করতে পারেন। এজন্য আগে থেকে স্টার আইকনের মতো কিছু আইকন অ্যাসাইন করে গ্রাফের আলাদা বারগুলো সিলেক্ট করে কলাম অপশনের মাধ্যমে নতুন আইকন অ্যাসাইন করা যাবে।



ইলাস্ট্রেটর ড্রইং করার জন্য খুব উন্নতমানের একটি সফটওয়্যার। এর মাধ্যমে ইউজার নিজের ইচ্ছে মতো ড্রইং করতে পারেন। তবে এই সফটওয়্যারের ড্রইংয়ের মূল মাধ্যম হলো ফটোশপের পেন টুল। ইলাস্ট্রেটরে ভালোভাবে ড্রইং করতে হলে পেন টুলের ব্যবহার ভালোভাবে শিখতে হবে **কল**

ফিডব্যাক : [wahid\\_cseast@yahoo.com](mailto:wahid_cseast@yahoo.com)