WELL COME

Programming in Java

Computer Science & Technology

4th Semester.

Presented by:

Md. Rabiul Awal
Chief Instructor(Tech.) Computer
Rangpur Polytechnic Institute, Rangpur.

১. অপারেটর কি ?

২. কন্ডিশনাল অপারেটর বলতে কি বুঝায়?

৩. বিটওয়াইজ অপারেটর কি?

এ ক্লাশ শেষে শিক্ষার্থীরা------

১. কন্ট্রোল ফ্লো স্টেটমেন্ট কি এবং কেন ব্যবহার করা হয় ?

২. বিভিন্ন প্রকার কন্ট্রোল ফ্লো স্টেটমেন্ট ব্যবহার করে জাভা প্রোগ্রাম লিখতে ও নির্বাহ করতে পারবে।

কন্ট্রোল স্টেটমেন্ট

জাভা প্রোগ্রামে কন্ট্রোল স্টেটমেন্ট ব্যবহার করা হয় মূলত: প্রোগ্রামের এক্সকিউশন ফ্লোকে নিয়ন্ত্রণ করার জন্য। কন্ট্রোল স্টেটমেন্ট সমূহ এক বা একাধিক ব্লকের এক্সকিউশন কন্ট্রোল করা হয়। জাভা প্রোগ্রামিং এ তিন প্রকার কন্ট্রোল স্টেটমেন্ট ব্যবহার করা হয়। যথা-

১) কন্ডিশনাল স্টেটমেন্ট।
২) লুপ স্টেটমেন্ট এবং
৩) আন-কন্ডিশনাল স্টেটমেন্ট।

১) কন্ডিশনাল স্টেটমেন্ট ঃ কন্ডিশনাল স্টেটমেন্ট হলো এমন এক ধরনের স্টেটমেন্ট যা একটি শর্ত পরীক্ষা করে এবং শর্ত অনুযায়ী নির্দিষ্ট কোড ব্লক সম্পাদিত করে। জাভাতে কন্ডিশনাল স্টেটমেন্ট সমূহ নিম্নরুপ-

ক) if স্টেটমেন্ট।
খ) if-else স্টেটমেন্ট।
গ) else if স্টেটমেন্ট।
ঘ) switch স্টেটমেন্ট।

২) লুপ স্টেটমেন্ট ঃ জাভা প্রোগ্রামিং ভাষায় লুপ স্টেটমেন্ট সমূহ ব্যবহার করে একই কার্যকে পুনরাবৃত্তি করার জন্য ব্যবহার করা হয়। লুপ স্টেটমেন্ট ব্যবহার করে একটি নির্দিষ্ট শর্ত পুরণ হলে একাধিক সংখ্যক বার এক্সিকিউশন চালানো যায়। জাভা ভাষায় লুপ স্টেটমেন্ট সমূহ নিম্নরুপঃ

ক) for লুপ।
খ) while লুপ এবং
গ) do-while লুপ।

৩) আন-কন্ডিশনাল স্টেটমেন্ট ঃ জাভা প্রোগ্রামিং এ আন-কন্ডিশনাল স্টেটমেন্ট হলো সেসব স্টেটমেন্ট যা কোন শর্তের অধীনে থাকে না বা শর্ত পরীক্ষা ছাড়াই সরাসরি এক্সিকিউশন করে। আন-কন্ডিশনাল স্টেটমেন্ট সমূহ নিম্নরুপঃ

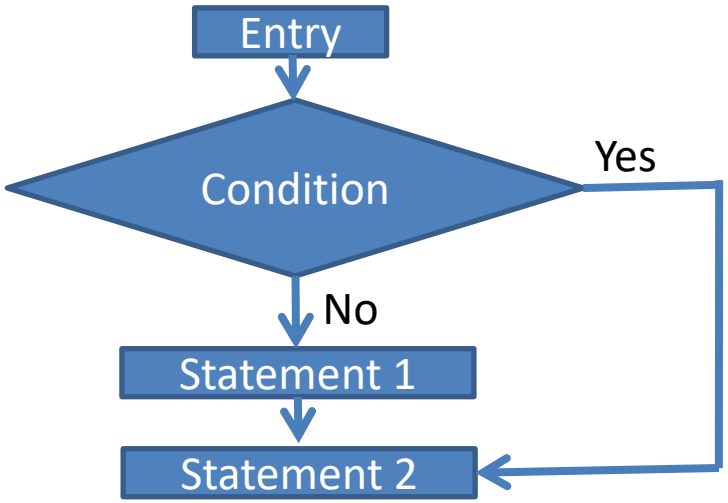
ক) break স্টেটমেন্ট।
খ) continue স্টেটমেন্ট।
গ) return স্টেটমেন্ট।
ঘ) throw স্টেটমেন্ট এবং
ঙ) try-catch-finally স্টেটমেন্ট।

If স্টেসমেন্ট এর সিনটেক্স ও ফ্লো-চার্ট

এই স্টেটমেন্টটি দ্বারা শর্ত সাপেক্ষে প্রোগ্রামের কোনো একটি লাইন বা একাধিক লাইনকে নির্বাহ করা যায় কিংবা নির্বাহ করা থেকে বিরত রাখা যায়। এখানে একটি শর্তের আওতায় একটি কাজ উল্লেখ করা থাকে, যদি শর্তটি সত্য হয় তবে কাজটি সম্পাদিত হয় অন্যথায় সম্পাদিত হয় না। নিম্নে স্টেটমেন্টটির সাধারন ফরম্যাট দেয়া হলো-

```
if(condition/test-expression):
        statement
```

ফ্লো-চার্টটি নিম্নে দেয়া হলোঃ

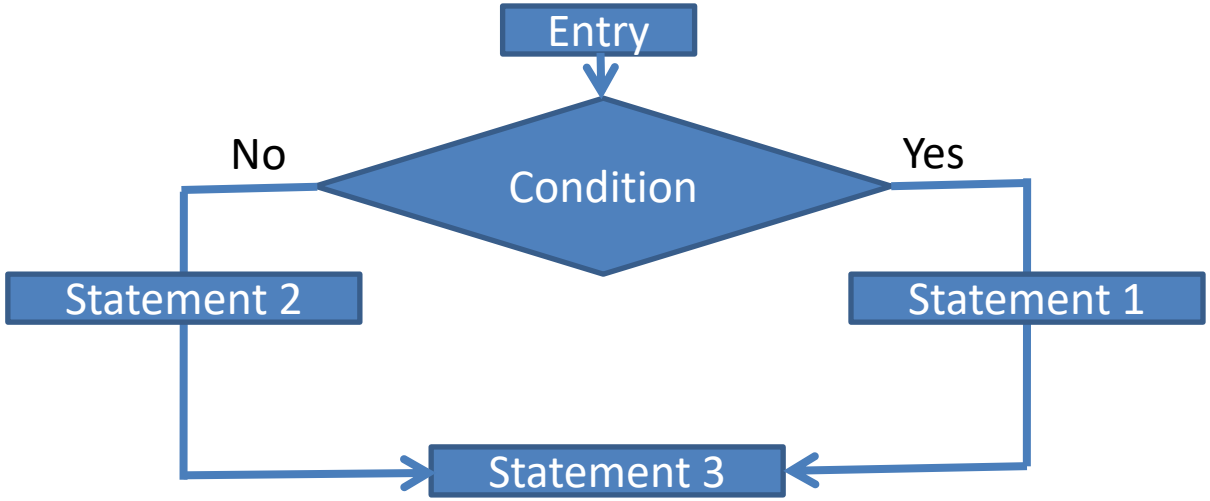


If-else স্টেসমেন্ট এর সিনটেক্স ও ফ্লো-চার্ট

এই স্টেটমেন্টটি দ্বারা শর্ত সাপেক্ষে প্রোগ্রামের কোনো একটি লাইন বা একাধিক লাইনকে নির্বাহ করা যায় । এখানে একটি শর্তের আওতায় একটি কাজ এবং বিকল্প একটি কাজ উল্লেখ করা থাকে, যদি শর্তটি সত্য হয় তবে কাজটি সম্পাদিত হয় অন্যথায় বিকল্প কাজটি সম্পাদিত হয়। নিম্নে স্টেটমেন্টটির সাধারন ফরম্যাট দেয়া হলো-

```
if(condition/test-expression)
        statement
else
        statement
```

ফ্লো-চার্টটি নিম্নে দেয়া হলোঃ



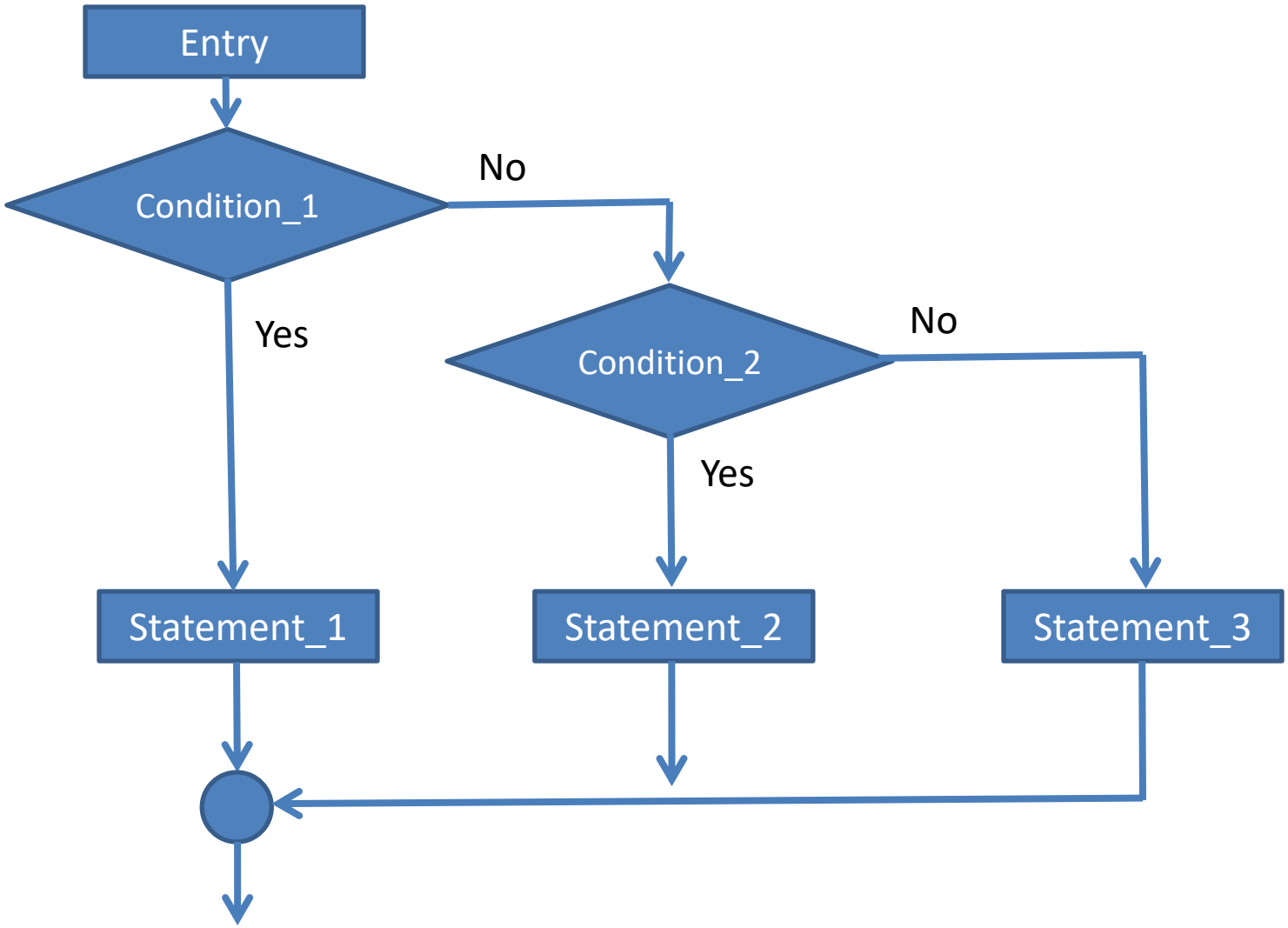
else if স্টেসমেন্ট এর সিনটেক্স ও ফ্লো-চার্ট

আমাদের যদি এমন কোন কাজ সমাধা করতে হয় যেখানে একাধিক শর্ত থাকবে এবং প্রত্যেকটি শর্তের আওতায় আলাদা আলাদা কাজ করতে হয়, তবে সেক্ষেত্রে elif স্টেটমেন্ট ব্যবহার করা হয়। এখানে একাধিক শর্তের মধ্যে যে শর্তটি সত্য হবে কেবলমাত্র তার আওতাধীন কোড সমূহ নির্বাহ হবে।

else if এর সাধারণ ফরম্যাট নিম্নরুপঃ

```
if(condition1)
    statement_1
else if(condition2)
    statement_2
else if(condition3)
    statement_3
---
---
else:
    statement_n
```

else if স্টেসমেন্ট এর ফ্লো-চার্ট



switch স্টেসমেন্ট এর সিনটেক্স

else if স্টেসমেন্ট এর অনুরুপ কাজে অর্থাৎ একাধিক স্টেটমেন্ট থেকে নিদির্ষ্ট কোনো স্টেটমেন্ট সম্পাদনের জন্য switch স্টেটমেন্ট ব্যবহার করা হয়। মূলতঃ বেশি সংখ্যক else if স্টেটমেন্ট ব্যবহারের পরিবর্তে switch স্টেটমেন্ট ব্যবহার করা হয় । switch এর ফরম্যাট নিম্নরুপঃ

```
data_type val;
switch(val)
{
case val1:
            {
            statements
            break;
            }
case val2:
            {
            statements
            break;
            }
default:
            {
            statement;
            break
            }
}
```

Loop স্টেসমেন্ট

জাভা প্রোগ্রামে কোন স্টেসমেন্ট দুই বা ততোধিক বার সম্পাদনের জন্য লুপ কন্ট্রোল স্টেটমেন্ট ব্যবহার করা হয়। লুপ স্টেটমেন্টের দুইটি অংশ থাকে। যথা-

ক) লুপ বডি এবং

খ) টেস্ট কন্ডিশন।

জাভা প্রোগ্রামে লুপ নির্বাহের জন্য **ব্যবহৃত অন্যতম লুপ কন্ট্রোল স্টেটমেন্ট সমূহ নিম্নরুপ-**

১) for স্টেটমেন্ট।

২) while স্টেটমেন্ট।

৩) do while স্টেটমেন্ট এবং

৪) continue স্টেটমেন্ট।

১ for স্টেটমেন্ট ঃ জাভা প্রোগ্রামিং এ কোন স্টেটমেন্ট দুই বা ততোধিক বার সম্পাদনের জন্য for স্টেটমেন্ট ব্যবহার করা হয়। এখানে একটি ভেরিয়েবল ব্যবহার করে লুপের আবর্তন সংখ্যা গণনা করা হয়। এ লুপের সাধারন ফরম্যাট নিম্নরুপঃ

```
for( Counter_Initialization;Condition;Increment/Decrement)
{
Loop Body
}
```

২ while স্টেটমেন্ট ঃ জাভা প্রোগ্রামিং এ কোন স্টেটমেন্টকে শর্তসাপেক্ষে দুই বা ততোধিক বার সম্পাদনের জন্য **while** স্টেটমেন্ট ব্যবহার করা হয়। এখানে একটি ভেরিয়েবল ব্যবহার করে লুপের আবর্তন সংখ্যা গণনা করা হয়। এ লুপের সাধারন ফরম্যাট নিম্নরুপঃ

```
while(Condition)
{
Loop Body
}
```

৩ do-while স্টেটমেন্ট ঃ জাভা প্রোগ্রামিং **while** এর বিকল্প হিসাবে **do** লুপ ব্যবহার করা হয়। **while** এর ক্ষেত্রে শর্ত প্রথমে পরীক্ষা করে, যদি শর্তটি সত্য হয় তবে স্টেটমেন্টটি সম্পাদিত হয়। কিন্তু **do** লুপের ক্ষেত্রে লুপ বডি একবার সম্পাদিত হয় এবং অত:পর শর্ত পরীক্ষা করে। এখানে শর্ত মিথ্যা হলেও বডি একবার সম্পাদিত হয়। শর্তটি দেয়া হয় লুপ বডির শেষে। এ লুপের সাধারন ফরম্যাট নিম্নরুপ ঃ

```
do
{
Loop Body
} while(Condition);
```

৪. continue স্টেটমেন্টঃ জাভা প্রোগ্রামে শর্তযুক্ত অথবা শর্তহীনভাবে কোন স্টেটমেন্ট বা লুপের পুনরাবৃত্তি করার জন্য এ স্টেটমেন্ট ব্যবহার করা হয়। continue স্টেটমেন্ট এর সাধারন ফরম্যাট নিম্নরুপঃ

```
continue;
```

try-catch-finally স্টেসমেন্ট

জাভা প্রোগ্রামিং এ try- catch-finally স্টেটমেন্টটি একটি এক্সেপশন হ্যান্ডেলিং মেকানিজম পরিচালনার জন্য ব্যবহার করা হয়। এ স্টেটমেন্টটি ব্যবহার করার ফলে প্রোগ্রামের নির্দিষ্ট অংশের এক্সেপশনের জন্য সম্পুর্ন প্রোগ্রামটি বন্ধ না হওয়া থেকে রক্ষা করে এবং প্রোগ্রামের স্বাভাবিক কার্যক্রম চলমান থাকে।

try- catch-finally ব্লকটি সিনট্যাক্স নিম্নরুপ ঃ

```
try{
// ট্রাই ব্লকে কোড রাখুন যা ত্রুটি উৎপন্ন করতে পারে।
}catch(Exception Type1 Exception-1){
// ত্রুটি দুর করার জন্য কোড সমূহ।
} catch(Exception Type1 Exception-1){
// ত্রুটি দুর করার জন্য কোড সমূহ।
} finally{
// এই ব্লক সব সময়ই সম্পাদিত হয়, এখানে রিসোর্স মুক্ত করণ এবং পরিস্কারকরণ কার্যক্রম পরিচালনা করা যায়।
}
```

অপারেটর ব্যবহার করে কিছু গুরুত্বপূর্ণ প্রোগ্রাম

দুইটি সংখ্যার মধ্যে সোয়াপিং করার একটি জাভা প্রোগ্রাম

```
import java.io.*;
public class Swap{
public static void main(String args[])throws Exception{
    BufferedReader br=new BufferedReader
     (new InputStreamReader(System.in));
    System.out.println("Enter first number");
    String n=br.readLine();
     int a=Integer.parseInt(n);
     System.out.println("Enter 2nd number");
    n=br.readLine();
     int b=Integer.parseInt(n);
     System.out.println("Before Swaping a= "+a +" & b= "+b);
     a=a+b;
     b=a-b;
     a=a-b;
     System.out.println("After Swaping a= "+a +" & b= "+b);
                        }
                                                          }
```

Output :

```
Enter first number
10
Enter 2nd number
20
Before Swaping a= 10 & b= 20
After Swaping a= 20 & b= 10
```

শর্ত সাপেক্ষে ত্রিভুজের ক্ষেত্রফল নির্ণয়ের প্রোগ্রাম

```
import java.io.*;
public class Triangle{
public static void main(String args[])throws Exception{
   BufferedReader br=new BufferedReader
     (new InputStreamReader(System.in));
   System.out.print("Enter the value of a :");
   String n=br.readLine();
     int a=Integer.parseInt(n);
     System.out.print("Enter the value of b :");
   n=br.readLine();
     int b=Integer.parseInt(n);
     System.out.print("Enter the value of c :");
   n=br.readLine();
     int c=Integer.parseInt(n);
```

```
    if(a+b>c && b+c>a && c+a>b)
    {
        double s=(a+b+c)/2.00;
        double area=Math.sqrt(s*(s-a)*(s-b)*(s-c));
        System.out.println(" Area of the Triangle is : "+area);
    }
    else
    {
        System.out.println("Triangle is not Possible");
    }
}
}
```

Output :

```
Enter the value of a :2
Enter the value of b :3
Enter the value of c :4
Area of the Triangle is :
  2.9047375096555625
```

তিনটি সংখ্যার মধ্যে ছোট সংখ্যাটি বের করার প্রোগ্রাম

```
import java.io.*;
public class Small{
public static void main(String args[])throws Exception{
   BufferedReader br=new BufferedReader
     (new InputStreamReader(System.in));
   System.out.print("Enter first value :");
   String n=br.readLine();
     int a=Integer.parseInt(n);
     System.out.print("Enter second value :");
   n=br.readLine();
     int b=Integer.parseInt(n);
     System.out.print("Enter third value :");
   n=br.readLine();
     int c=Integer.parseInt(n);
     int small;
     if(a<b)
     {
            if(a<c)
                        {small=a;}
            else
                        {small=c;}
     }
```

```
    else
    {
            if(b<c)
                    {small=b;}
            else
                    {small=c;}
    }
    System.out.println(" The small number of "+a +" ,"+b+" & "+c +" is  :
    "+ small);
}
}
```

Output :

```
Enter first value :50
Enter second value :45
Enter third value :3
 The small number of 50 ,45 & 3 is  : 3
```

Switch স্টেসমেন্ট ব্যবহার করে গ্রেড বের করার প্রোগ্রাম

```
import java.io.*;
public class Grade{
public static void main(String args[])throws Exception{
   BufferedReader br=new BufferedReader
    (new InputStreamReader(System.in));
   System.out.print("Enter a mark between 1 to 10 :");
   String n=br.readLine();
    int a=Integer.parseInt(n);
    String gra;
    switch(a)
    {
            case 10:
            case 9:
            case 8:
                        gra="A+";
                        break;
            case 7:
                        gra="A";
                        break;
            case 6:
                        gra="B";
                        break;
```

```
            case 5:
                        gra="C";
                        break;
            case 4:
                        gra="D";
                        break;
            default:
                        gra="F";
                        break;

    }
    System.out.println(" Your grade is :"+gra);
}
}
```

Output :

```
Enter a mark between 1 to 10 :5
Your grade is :C
```

20,18,16,- - - ,06 এই সিরিজের যোগফল বের করার প্রোগ্রাম

```
import java.io.*;
public class Sum1{
public static void main(String args[])
    {
    int sum=0;
    for(int i=20;i>=06;i-=2)
            {
            sum=sum+i;
            }
    System.out.println(" Sum of the Series is :"+sum);
    }
}
```

Output :

```
Sum of the Series is :104
```

ফিবোনসি সিরিজ প্রদর্শনের প্রোগ্রাম

```
import java.io.*;
public class Fibo{
public static void main(String args[])throws Exception{
   BufferedReader br=new BufferedReader
     (new InputStreamReader(System.in));
   System.out.print("Enter how many number in the series :");
   String n=br.readLine();
     int a=Integer.parseInt(n);
     int f,f0=1,f1=2;
     System.out.print(f0+"  "+f1);
     for(int i=2;i<=a-1;i++)
     {
            f=f0+f1;
            f0=f1;f1=f;
            System.out.print("  "+f);
     }
}
}
```

Output :

```
Enter how many number in the series :10
1  2  3  5  8  13  21  34  55  89
```

do while loop ব্যবহার করে ০ থেকে ১০০ পর্যন্ত ৫ দ্বারা বিভাজ্য সংখ্যার যোগফল বের করার প্রোগ্রাম

```
import java.io.*;
public class Sum2{
public static void main(String args[])
    {
    int sum=0, i=0;
    do
        {
        sum=sum+i;
        i+=5;
        }while(i<=100);
    System.out.println(" Sum of the Series is :"+sum);
    }
}
```

Output :

```
Sum of the Series is :1050
```

১ থেকে ১০০ পর্যন্ত মৌলিক সংখ্যাগুলো প্রিন্ট করার প্রোগ্রাম

```
import java.io.*;
public class PrimeTest{
public static void main(String args[])
    {
    for( int i=2;i<=100;i++)
            {           int pt=1;
                        for(int j=2;j<=i;j++)
                        {
                                    if(i % j==0 && j!=i)
                                    {
                                                pt=0;
                                    }
                        }
                        if(pt==1)
                        {
                                    System.out.print(" "+i);
                        }
            }
    }
}
```

Output :

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

১. কন্ডিশনাল কন্ট্রোল ফ্লো বলতে কি বুঝায়?

2. while ও do-while লুপের মধ্যে পার্থক্য কি?

পরবর্তী ক্লাশে আলোচনা করা হবে

অ্যারে

Thanks to All