
লুপ

লেকচার-১৬

লুপ

লেকচার-১৬

এই পাঠ শেষে যা যা শিখতে পারবে-

- ১। লুপ ব্যাখ্যা করতে পারবে।
- ২। লুপ কন্ট্রোল স্টেটমেন্ট ব্যাখ্যা করতে পারবে।
- ৩। for লুপ স্টেটমেন্ট, while লুপ স্টেটমেন্ট ও do...while লুপ স্টেটমেন্ট ব্যাখ্যা করতে পারবে।
- ৪। while লুপ স্টেটমেন্ট ও do while লুপ স্টেটমেন্টের মধ্যে পার্থক্য করতে পারবে।

লুপ

প্রোগ্রামের এক বা একাধিক স্টেটমেন্ট নির্দিষ্ট সংখ্যক বার পুনরাবৃত্তি করাকে লুপ বা লুপিং বলে।

লুপের প্রকারভেদ:

সসীম লুপ - যদি কোন লুপ নির্দিষ্ট সংখ্যক বার পুনরাবৃত্তি হয়, তখন তাকে সসীম লুপ বলে।

অসীম লুপ - যদি কোন লুপ অনবরত পুনরাবৃত্তি হতে থাকে, অর্থাৎ কখনো শেষ না হয় তবে তাকে অসীম লুপ বলে।

মধ্যবর্তী লুপ - একটি লুপের মধ্যে যদি অপর একটি লুপ থাকে তাহলে তাকে মধ্যবর্তী বা ন্যেস্টেড লুপ বলে।

লুপ কন্ট্রোল স্টেটমেন্ট

প্রোগ্রামের এক বা একাধিক স্টেটমেন্ট নির্দিষ্ট সংখ্যক বার পুনরাবৃত্তি করার জন্য যে কন্ট্রোল স্টেটমেন্ট ব্যবহৃত হয় তাকে লুপ কন্ট্রোল স্টেটমেন্ট বলে।

লুপ কন্ট্রোল স্টেটমেন্ট সমূহ

- ১। for লুপ স্টেটমেন্ট
- ২। while লুপ স্টেটমেন্ট
- ৩। do...while লুপ স্টেটমেন্ট

প্রত্যেক লুপ কন্ট্রোল স্টেটমেন্টের দুটি অংশ থাকে।
যথা-

- ১। লুপ ডিক্লারেশন
- ২। লুপ বডি

লুপ ডিক্লারেশন তিনটি প্রধান অংশ-

- ১। Initialization Statement- এই অংশে ভেরিয়েবলের প্রাথমিক মান নির্ধারণ করা হয়।
- ২। Test Expression - এই অংশে শর্ত লেখা হয়। শর্ত মিথ্যা না হওয়া পর্যন্ত লুপ বডি পুনরাবৃত্তি হয়।
- ৩। Update Statement- প্রতিবার লুপ বডি পুনরাবৃত্তির পর ভেরিয়েবলের মান হ্রাস/বৃদ্ধি নির্ধারণ করা হয় এই অংশে।

লুপ বডি- যে স্টেটমেন্টগুলো পুনরাবৃত্তি হবে তা { } এর মধ্যে থাকে, যা লুপ বডি হিসেবে বিবেচিত হয়।

থাকলেই কেবলমাত্র ভড়ৎ লুপ ব্যবহার করা যায়। নিম্নে for লুপ স্টেটমেন্টের ফরম্যাট দেওয়া হলো-

```
for (initializationStatement; testExpression; updateStatement)
{
    // codes
}
```

লুপ স্টেটমেন্টের লুপ বডি এবং টেস্ট কন্ডিশনের অবস্থানের ভিত্তিতে লুপ স্টেটমেন্টসমূহকে দুই ভাগে ভাগ করা যায়। যথা-

১। এন্ট্রি কন্ট্রোল লুপ স্টেটমেন্ট

২। এক্সিট কন্ট্রোল লুপ স্টেটমেন্ট

এন্ট্রি কন্ট্রোল লুপ স্টেটমেন্টঃ লুপ বডি নির্বাহের পূর্বে টেস্ট কন্ডিশন যাচাই করা হয়। কন্ডিশন সত্য হলেই কেবলমাত্র লুপ বডি নির্বাহ হয়। উদাহরণঃ for লুপ স্টেটমেন্ট, while loop স্টেটমেন্ট।

এক্সিট কন্ট্রোল লুপ স্টেটমেন্টঃ প্রথমবার টেস্ট কন্ডিশন যাচাই না করেই লুপ বডি নির্বাহ হয়। তারপর কন্ডিশন যাচাই করা হয়। কন্ডিশন সত্য হলে লুপ বডি নির্বাহ হয়। উদাহরণঃ do-while loop স্টেটমেন্ট।

for loop যেভাবে কাজ করে-

১। প্রথমে Initialization Statement নির্বাহ হয়। লুপ স্টেটমেন্টে Initialization Statement কেবলমাত্র একবার নির্বাহ হয়।

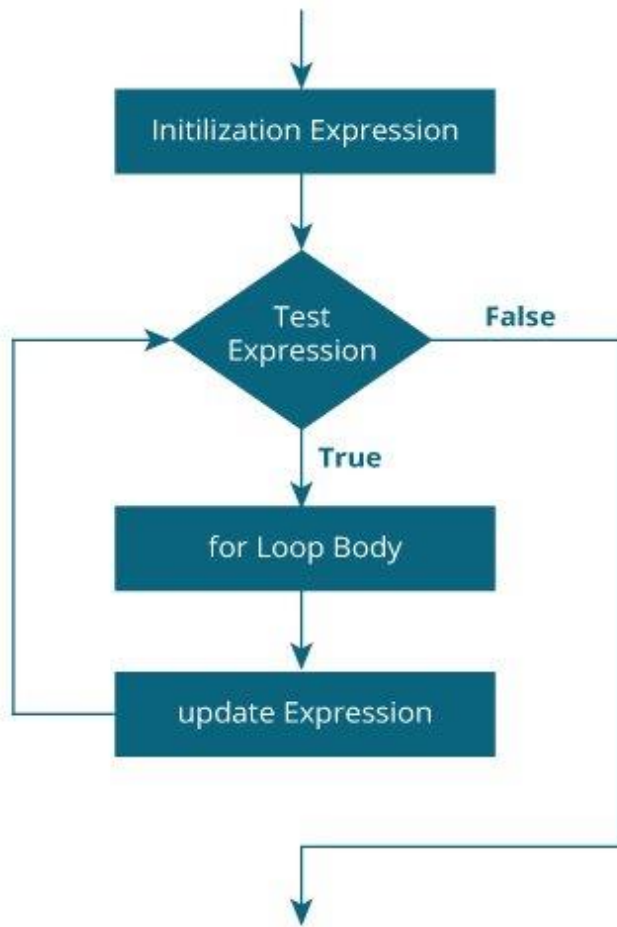
২। তারপর Test Expression চেক করে। যদি Test Expression সত্য হয় তাহলে লুপ বডির কোড নির্বাহ হয় এবং Update Expression এর মান আপডেট হয়। Test Expression মিথ্যা না হওয়া পর্যন্ত ধাপ-২ পুনরাবৃত্তি হতে থাকে।

৩। যদি Test Expression মিথ্যা হয়, তাহলে প্রোগ্রাম নির্বাহ লুপ থেকে বের হয়ে আসে।

for লুপ স্টেটমেন্টঃ

‘সি’ প্রোগ্রামে শর্ত সাপেক্ষে এক বা একাধিক স্টেটমেন্ট নির্দিষ্ট সংখ্যকবার নির্বাহ করতে ভড়ৎ লুপ স্টেটমেন্ট ব্যবহার করা হয়। লুপ বডির কোড নির্বাহের পূর্বে কন্ডিশন চেক করে। লুপ কতবার নির্বাহ হবে তা জানা

for লুপের ফ্লোচার্ট



while loop স্টেটমেন্ট

‘সি’ প্রোগ্রামে শর্ত সাপেক্ষে এক বা একাধিক স্টেটমেন্ট নির্দিষ্ট সংখ্যকবার নির্বাহ করতে while loop স্টেটমেন্ট ব্যবহার করা হয়। লুপ বডির কোড নির্বাহের পূর্বে কন্ডিশন চেক করে while loop কে for loop এর বিকল্প হিসাবে ব্যবহার করা যায়। লুপ কতবার নির্বাহ হবে তা অজানা থাকলে যিরষব লুপ ব্যবহার করা হয়। while loop স্টেটমেন্টের ফরম্যাট হলো-

```

while (testExpression)
{
    //codes
}
  
```

এবারে আমরা ভড়ৎ ষড়্‌ড়ৎ ব্যবহার করে ঐবষষড়্‌ ডড়ৎষফ লেখাটি ৫ বার প্রিন্ট করার প্রোগ্রাম তৈরি করব।

```

#include<stdio.h>
#include<conio.h>

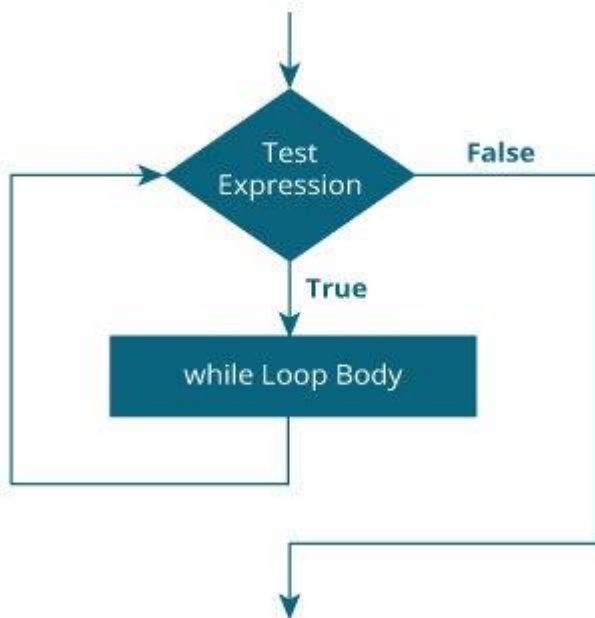
main()
{
    int i;
    for(i=1; i<=5; i++)
    {
        printf("Hello World\n");
    }
    getch();
}
  
```

while loop যেভাবে কাজ করে-

১। প্রথমে Test Expression চেক করে। যদি Test Expression সত্য হয় তাহলে লুপ বডির কোড নির্বাহ হয় এবং পুনরায় Test Expression চেক করে। Test Expression মিথ্যা না হওয়া পর্যন্ত এই প্রক্রিয়া পুনরাবৃত্তি হতে থাকে।

২। যদি Test Expression মিথ্যা হয়, তাহলে প্রোগ্রাম নির্বাহ লুপ থেকে বের হয়ে আসে।

while লুপের ফ্লোচার্টঃ



এবারে আমরা While loop ব্যবহার করে Hello World লেখাটি ৫ বার প্রিন্ট করার প্রোগ্রাম তৈরি করব।

```
#include<stdio.h>
#include<conio.h>

main()
{
    int i;
    i=1;
    while(i<=5)
    {
        printf("Hello World\n");
        i++;
    }
    getch();
}
```

for লুপের মত করে While লুপের গঠনঃ

```
initializationStatement;
while(testExpression)
{
    // codes;

    updateStatement;
}
```

do-while loop স্টেটমেন্টঃ

‘সি’ প্রোগ্রামে শর্ত সাপেক্ষে এক বা একাধিক স্টেটমেন্ট নির্দিষ্ট সংখ্যকবার নির্বাহ করতে do-while loop স্টেটমেন্ট ব্যবহার করা হয়। do-while লুপের কন্ডিশন যাচাই না করে লুপ বডি অন্ততপক্ষে একবার নির্বাহ হয়। কারণ এখানে কন্ডিশন পরে যাচাই হয়। do-while loop টি do loop নামেও পরিচিত। তবে প্রোগ্রামে ভড়ৎ এবং while লুপের চেয়ে do-while loop লুপ কম ব্যবহৃত হয়। do-while loop এর গঠন হচ্ছে-

```
do
{
    // codes
}
while (testExpression);
```

for লুপের মত করে do-while লুপের গঠনঃ

```
initializationStatement;
do
{
    // codes;

    updateStatement;
}while(testExpression);
```

do-while loop যেভাবে কাজ করে-

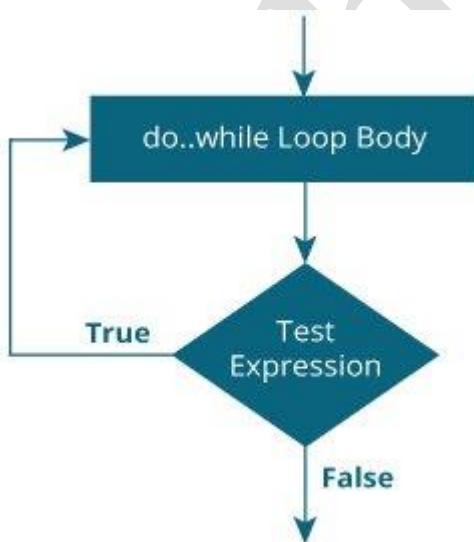
- ১। প্রথমে লুপ বডি কোড একবার নির্বাহ হয়।
- ২। তারপর Test Expression চেক করে। যদি Test Expression সত্য হয় তাহলে লুপ বডি কোড নির্বাহ হয় এবং পুনরায় Test Expression চেক করে। Test Expression মিথ্যা না হওয়া পর্যন্ত এই প্রক্রিয়া পুনরাবৃত্তি হতে থাকে।
- ৩। যদি Test Expression মিথ্যা হয়, তাহলে প্রোগ্রাম নির্বাহ লুপ থেকে বের হয়ে আসে।

এবারে আমরা do-while loop ব্যবহার করে Hello World লেখাটি ৫ বার প্রিন্ট করার প্রোগ্রাম তৈরি করব।

```
#include<stdio.h>
#include<conio.h>

main()
{
    int i;
    i=1;
    do
    {
        printf("Hello World\n");
        i++;
    }while(i<=5);
    getch();
}
```

do-while লুপের ফ্লোচার্টঃ



পাঠ মূল্যায়ন-

জ্ঞানমূলক প্রশ্নসমূহ

১। লুপ কী?

উত্তরঃ প্রোগ্রামের অংশ বিশেষ কোনো শর্তে না পৌঁছা পর্যন্ত নির্দিষ্ট সংখ্যক বার পুনরাবৃত্তি করাকে লুপিং বা লুপ বলা হয়।

২। লুপ কন্ট্রোল স্টেটমেন্ট কী?

উত্তরঃ প্রোগ্রামের এক বা একাধিক স্টেটমেন্ট নির্দিষ্ট সংখ্যক বার পুনরাবৃত্তি করার জন্য যে কন্ট্রোল স্টেটমেন্ট ব্যবহৃত হয় তাকে লুপ কন্ট্রোল স্টেটমেন্ট বলে।

অনুধাবনমূলক প্রশ্নসমূহ

১। for লুপ এবং while লুপের মধ্যে কোনটি ব্যবহার করা সহজ?

২। while লুপ এবং do-while লুপের মধ্যে পার্থক্য আলোচনা কর।