PDP
IT-ACADEMY

Takrorlash operatorlari (2-qism)

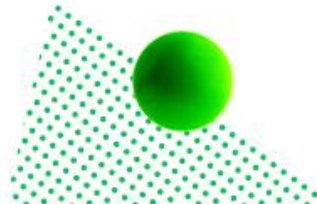
PDP
IT-ACADEMY

Reja:

- break
- continue
- return
- Nested loops
- Amaliy mashqlar

PDP
IT-ACADEMY

break



PDP
IT-ACADEMY

break operatori o’zi joylashgan takrorlash operatorining bajarilishini to’xtatadi.

PDP
IT-ACADEMY

```
for (init; condition; update) {
    // code
    if (condition to break) {
        break;
    }
    // code
}
```

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main() {
    for (int i = 1; i <= 5; i++) {
        // break condition
        if (i == 3) {
            break;
        }
        cout << i << endl;
    }

return 0;
}
```

```
1
2
```

PDP
IT-ACADEMY

```
while (condition) {
    // code
    if (condition to break) {
        break;
    }
    // code
}
```

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main() {
  int i = 0;

  while (i < 10) {
    cout << i << "\n";
    i++;
    if (i == 7) {
      break;
    }
  }

  return 0;
}
```

```
0
1
2
3
4
5
6
```

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main() {
    int i = 0;

    do {
        if (i == 7) {
            break;
        }
        cout << i << '\n';
        i++;
    } while (i < 10);
}
```

```
0
1
2
3
4
5
6
```

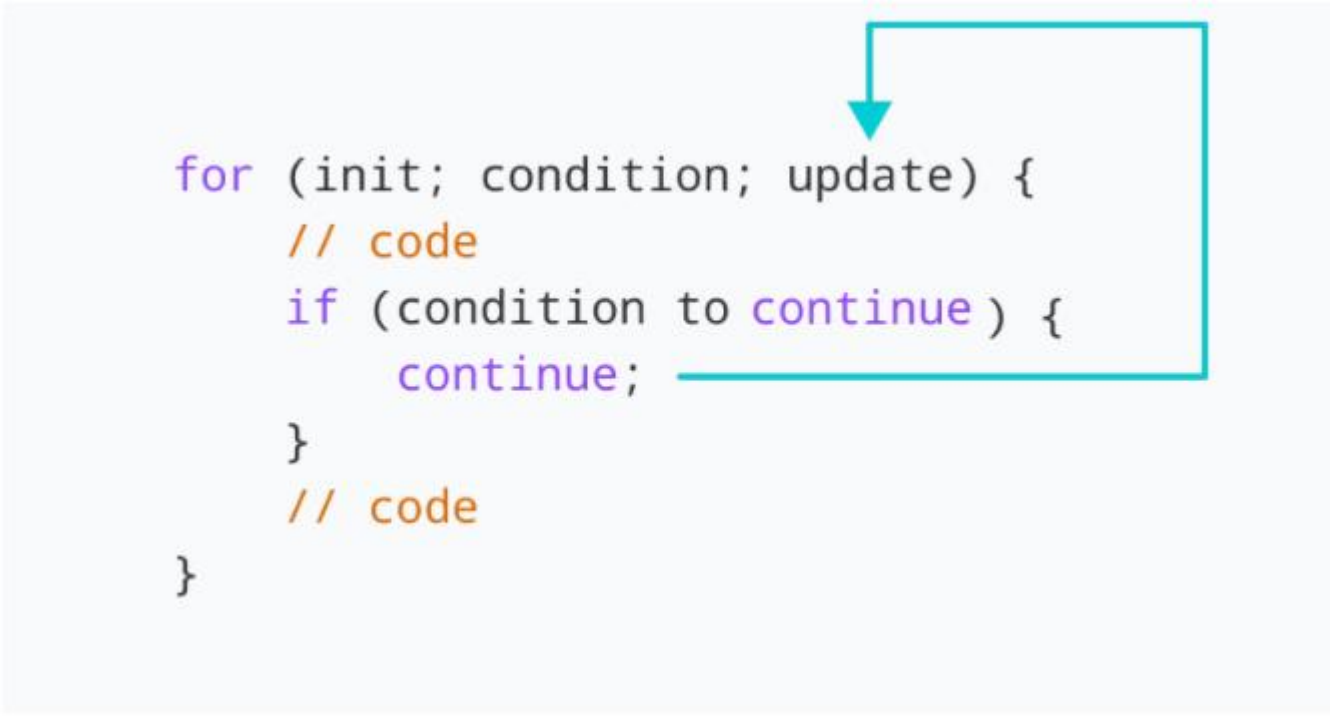
PDP
IT-ACADEMY

continue

PDP
IT-ACADEMY

continue operatori o’zi joylashgan takrorlash operatorining bajarilishini bir qadamga chetlab o’tadi, ya’ni bajarilishda o’zi uchragan joydan takrorlash tanasining oxirigacha bo’lgan operatorlarni bajarilmasligini ta’minlaydi.

PDP
IT-ACADEMY

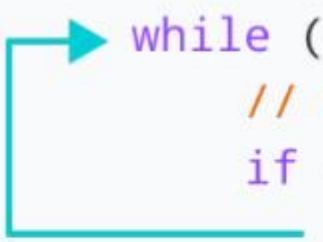


PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main() {
    for (int i = 1; i <= 5; i++) {
        // condition to continue
        if (i == 3) {
            continue;
        }

        cout << i << endl;
    }

    return 0;
}
```

```
1
2
4
5
```

PDP
IT-ACADEMY



```
while (condition) {
    // code
    if (condition to continue ) {
        continue;
    }
    // code
}
```

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main() {
  int i = 0;
  while (i < 10) {
    if (i == 4) {
      i++;
      continue;
    }
    cout << i << "\n";
    i++;
  }
  return 0;
}
```

```
0
1
2
3
5
6
7
8
9
```

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main () {
   int a = 10;

   do {
      if( a == 15) {
         a = a + 1;
         continue;
      }
      cout << "value of a: " << a << endl;
      a = a + 1;
   }
   while( a < 20 );

   return 0;
}
```

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

PDP
IT-ACADEMY

return

PDP
IT-ACADEMY

return operatori o’zi joylashgan funksiyaning bajarilishini to’xtatadi. Agar funksiya qiymat qaytarishi kerak bo’lsa, kerakli natijani qaytaruvchi operator hisoblanadi.

```
return [expression];
```

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main() {
  for (int i = 0; i < 10; i++) {
    if (i == 4) {
      break;
    }
    cout << i << " ";
  }

  cout<<"Dastur ishi yakunlandi"<<endl;
  return 0;
}
```

```
0 1 2 3 Dastur ishi yakunlandi
```

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main() {
  for (int i = 0; i < 10; i++) {
    if (i == 4) {
      return 0;
    }
    cout << i << " ";
  }

  cout<<"Dastur ishi yakunlandi"<<endl;
  return 0;
}
```

```
 0 1 2 3
```

PDP
IT-ACADEMY

Nested loops

PDP
IT-ACADEMY

```
for ( init; condition; updation ) {
    for ( init; condition; updation ) {
        statement(s);
    }
    statement(s);
}
```

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main() {
    int weeks = 3, days_in_week = 7;

    for (int i = 1; i <= weeks; ++i) {
        cout << "Week: " << i << endl;

        for (int j = 1; j <= days_in_week; ++j) {
            cout << "    Day:" << j << endl;
        }
    }

    return 0;
}
```

```
Week: 1
    Day:1
    Day:2
    Day:3
    Day:4
    Day:5
    Day:6
    Day:7
Week: 2
    Day:1
    Day:2
    Day:3
    Day:4
    Day:5
    Day:6
    Day:7
Week: 3
    Day:1
    Day:2
    Day:3
    Day:4
    Day:5
    Day:6
    Day:7
```

PDP
IT-ACADEMY

```
while(condition) {
    while(condition) {
        statement(s);
    }
    statement(s);
    // you can put more statements.
}
```

PDP
IT-ACADEMY

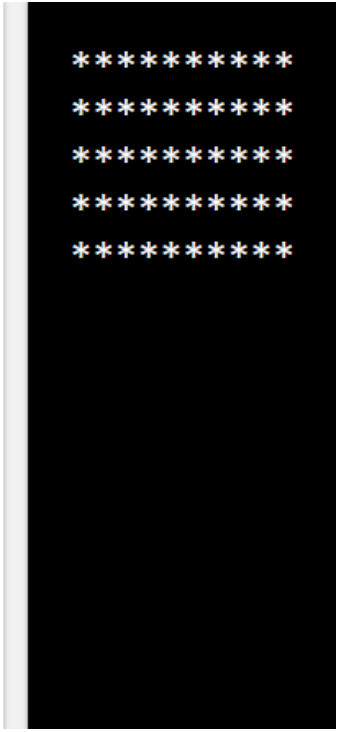
```
do {
   statement(s);
   // you can put more statements.
   do {
      statement(s);
   } while( condition );

} while( condition );
```

PDP
IT-ACADEMY

```
do{
   while(condition) {
      for ( initialization; condition; updation ) {
         // statement of inside for loop
      }
      // statement of inside while loop
   }
   // statement of outer do-while loop
}while(condition);
```

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main()
{
    // nested for loops
    for (int i = 0; i < 5; i++) {
        for (int j = 1; j <= 10; j++) {
                cout << "*";
        }
        cout << endl;
    }

    return 0;
}
```

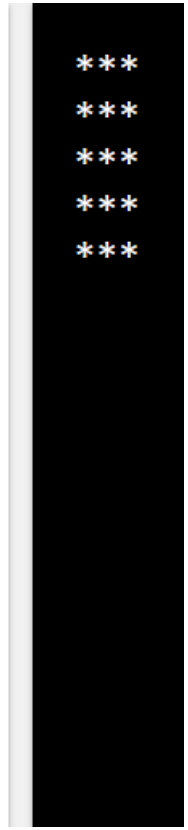


PDP
IT-ACADEMY

break in nested loops

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main()
{
    for (int i = 0; i < 5; i++) {
        for (int j = 1; j <= 10; j++) {
            if (j > 3)
                break;
            else
                cout << "*";
        }
        cout << endl;
    }

    return 0;
}
```



PDP
IT-ACADEMY

continue in nested loops

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main()
{
    for (int i = 0; i < 5; i++) {
        for (int j = 1; j <= 10; j++) {
            if (j > 3 && j < 7)
                continue;
            else
                cout << "*";
        }
        cout << endl;
    }

    return 0;
}
```

```
*******
*******
*******
*******
*******
```

PDP
IT-ACADEMY

return in nested loops

PDP
IT-ACADEMY

```cpp
#include <iostream>
using namespace std;

int main()
{
    for (int i = 0; i < 5; i++) {
        for (int j = 1; j <= 10; j++) {
            if (j > 3)
                return 0;
            else
                cout << "*";
        }
        cout << endl;
    }

    return 0;
}
```

PDP
IT-ACADEMY

Amaliy mashqlar

PDP
IT-ACADEMY

Foydalanuvchi tomonidan sonlar kiritilaveradi. Bu jarayon musbat bo’lmagan son kiritilguncha davom etadi. Kiritilgan musbat sonlarning yig’indisini toping.

PDP
IT-ACADEMY

Foydalanuvchi tomonidan sonlar kiritilaveradi. Bu jarayon musbat bo’lmagan son kiritilguncha davom etadi. Kiritilgan musbat sonlarning yig’indisini toping. Bunda yig’indiga 50 dan katta kiritilgan sonlar qo’shilmaydi.

PDP
IT-ACADEMY

N natural soni berilgan. Uning tub yoki tub emasligini aniqlovchi dastur tuzing.

Tub son – bu faqat o’ziga va 1 ga qoldiqsiz bo’linadigan son.

1 soni tub ham, murakkab ham emas.

PDP
IT-ACADEMY

Karra jadvalini to’liq (2 dan 9 gacha) ekranga chiqaring.

PDP
IT-ACADEMY

N natural soni berilgan. Unga mos ravishda quyidagi shaklni ekranga chiqaruvchi dastur tuzing.

```
Input: 7
Output:
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
```

PDP
IT-ACADEMY

E`tiboringiz uchun rahmat!