On Things Omitted

- For, do while
- Break, continue
- Variable scoping
- Switch
- ++,--,+=,*=,-=,/=

for

- for(i=1;i<b;i=i+1) statment
- i=1;
  while(i<b){
  statement
  i=i+1;
  }
- prod=1;
  for(i=1;i<=n;i=i+1) prod=prod*i;
  What does this compute?

for

- for(init;condition;increment) statment
- init;

```
while(condition){
   statement
  increment;
}
```

- Where init,condition,increment are any valid statements. But just choosing any valid statement does not mean it will do something useful.

for

- Examples
  - for(i=1,prod=1;i<=n;i=i+1) prod=prod*i;
  - for (int i=1;i>0;i==5;) i=i+1; // infinite loop
  - for (int i=0, j=n;i<j;i=i+1, j=n-2*i){ }
- Typically usage
  - for (int i=0; i<n;i=i+1){ }
  - for (int i=n; i>0;i=i-1){ }

Do While

- do{ statements } while (condition);
- test=true;

  while(test){

    statements

    test=condition;

  }
- Int prod=1,value=1;

  do {prod=prod*value;cin>>value}

    while(value >0);

break

- The break statement terminates the execution of the nearest enclosing **do**, **for**, **switch**, or **while** statement in which it appears. Control passes to the statement that follows the terminated statement.
- ```
  while(1){
    cin >>value
    if (value <1) break;
    else prod=prod*value;
  }
  ```
  ```

break

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

int main() {
   char c;
   for(;;) {
    cout << "\nPress any key, Q to quit: "<<
          endl;
    cin >> c;
   if (c == 'Q')
          break;
   }
   } // Loop exits only when 'Q' is pressed
```

continue

- The **continue** statement passes control to the next iteration of the nearest enclosing **do**, **for**, or **while** statement in which it appears, bypassing any remaining statements in the **do**, for, or **while** statement body.

```
#include <iostream>
using namespace std;
int main() {
  int n,prod,i;
  cout << "Enter a positive integer" << endl;
  cin >> n;
 for( i=1,prod=1;i<=n;i=i+1){
      if (i % 2 == 1) continue;
      prod=prod*i;
    }
    cout << "The product of the even numbers <= "<< n << " is "
        << prod << endl;
}
```

Variable Scope

- Memory locations declared in { } will not exist outside of the {}
-
```
#include <iostream>
using namespace std;
int main() {
  int n,prod=0,i;
  cout << "Enter a positive integer" << endl;
  cin >> n;
 for( i=1;i<=n;i=i+1){
      int prod;
      if (i==1) prod=1;
      if (i % 2 == 1) continue;
      prod=prod*i;
    }
    cout << "The product of the even numbers <= "<< n << " is "
        << prod << endl;
}
```

The Switch Statement

```
switch( c ) //count number of A's, A's and a's
// and total number of letters
{
    case 'A':   // if c=='A' execute from here until break or }
        capa++;
    case 'a':// if c=='a' execute from here until break or }
        lettera++;
    default :  execute from here until break or }
        total++;
}
```

The Switch Statement

```
switch( i )
{
    case -1:   // if i==-1 execute from here until break or }
        n=n+1;break;
    case 0:// if i==0 execute from here until break or }
        z=z+1;break;
    case 1:  if i==1 execute from here until break or }
        p=p+1;
}
```

Increment, decrement operators

- ++p,p++, pre,post increment operator, increments by 1
- --p,p--, pre, post decrement operator decrements by 1
- ```
  #include <iostream>
   using namespace std;
   int main() {
     int pre_pp=5,post_pp=5,pre_mm=5,post_mm=5;
     cout << "++pre_pp, pre_pp" <<endl;
     cout <<++pre_pp <<","<<pre_pp <<endl;
     cout << "post_pp++, post_pp" <<endl;
     cout << post_pp++<<"," <<post_pp <<endl;
     cout << "--pre_mm, pre_mm" <<endl;
     cout << --pre_mm<<"," << pre_mm <<endl;
     cout << "post_mm--, post_mm" <<endl;
     cout << post_mm--<<","<< post_mm <<endl;
   }
  ```
- Output

  ```
  ++pre_pp, pre_pp
  6,6
  post_pp++, post_pp
  5,6
  --pre_mm, pre_mm
  4,4
  post_mm--, post_mm
  5,4
  ```
  ```

Increment, decrement operators

- p += n; //p=p+n;
- p -= n; //p=p-n;
- p *= n; //p=p*n;
- p /= n; //p=p/n;
- p %= n; //p=p%n;
- p=4, n=2;

p += n; //p=6

p -= n; //p=2;

p *= n; //p=8;

p /= n; //p=2;

p %= n; //p=0;