



Programming Fundamentals

Aamina Batool

Type Conversion (Casting)

- Implicit type coercion: when value of one type is automatically changed to another type
- Cast operator provides explicit type conversion
- Use the following form:
 - `static_cast<dataTypeName> (expression)`



Type Casting

- `static_cast<int>(7.9)`
- `static_cast<int>(3.3)`



Type Casting

- `static_cast<double>(25)`
- `static_cast<double>(5 + 3)`
- `static_cast<double>(15) / 2`
- `static_cast<double>(15 / 2)`



Type Casting

- `static_cast<int>(7.8 + static_cast<double>(15) / 2)`
- `static_cast<int>(7.8 + static_cast<double>(15 / 2))`

EXAMPLE 2-9

Expression

Evaluates to

`static_cast<int>(7.9)`

7

`static_cast<int>(3.3)`

3

`static_cast<double>(25)`

25.0

`static_cast<double>(5 + 3)`

= `static_cast<double>(8)` = 8.0

`static_cast<double>(15) / 2`

= 15.0 / 2

(because `static_cast<double>(15)` = 15.0)

= 15.0 / 2.0 = 7.5

`static_cast<double>(15 / 2)`

= `static_cast<double>(7)` (because $15 / 2 = 7$)

= 7.0

`static_cast<int>(7.8 +`

`static_cast<double>(15) / 2)`

= `static_cast<int>(7.8 + 7.5)`

= `static_cast<int>(15.3)`

= 15

`static_cast<int>(7.8 +`

`static_cast<double>(15 / 2))`

= `static_cast<int>(7.8 + 7.0)`

= `static_cast<int>(14.8)`

= 14



Using namespace std

- `std :: cout`
- `std :: cin`
- `std :: endl`



Repetition

- ➡ Print first 5 natural numbers



Repetition

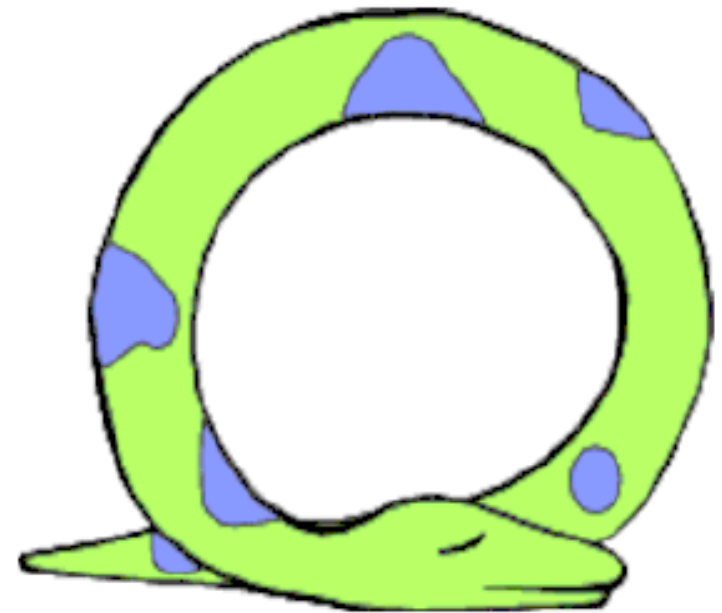
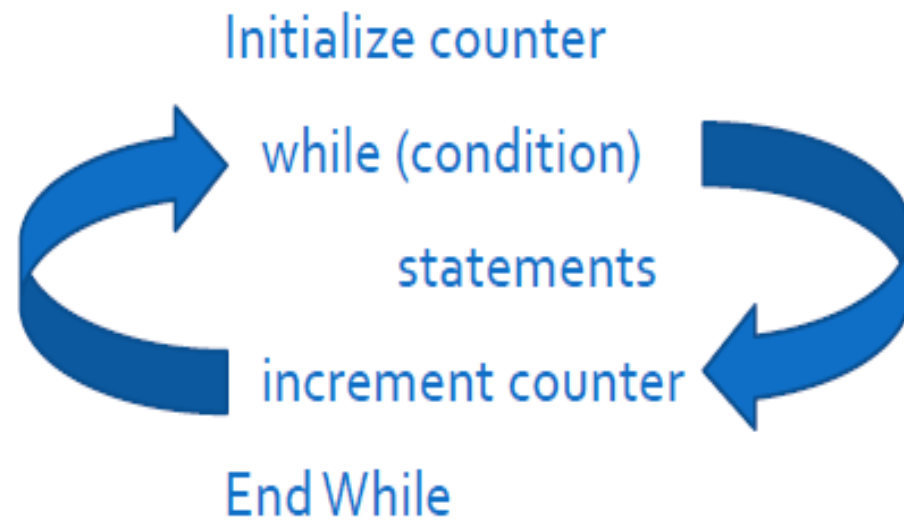
- ➡ Print first 15 natural numbers



Repetition

- ➡ Print first 1000 natural numbers

Loop



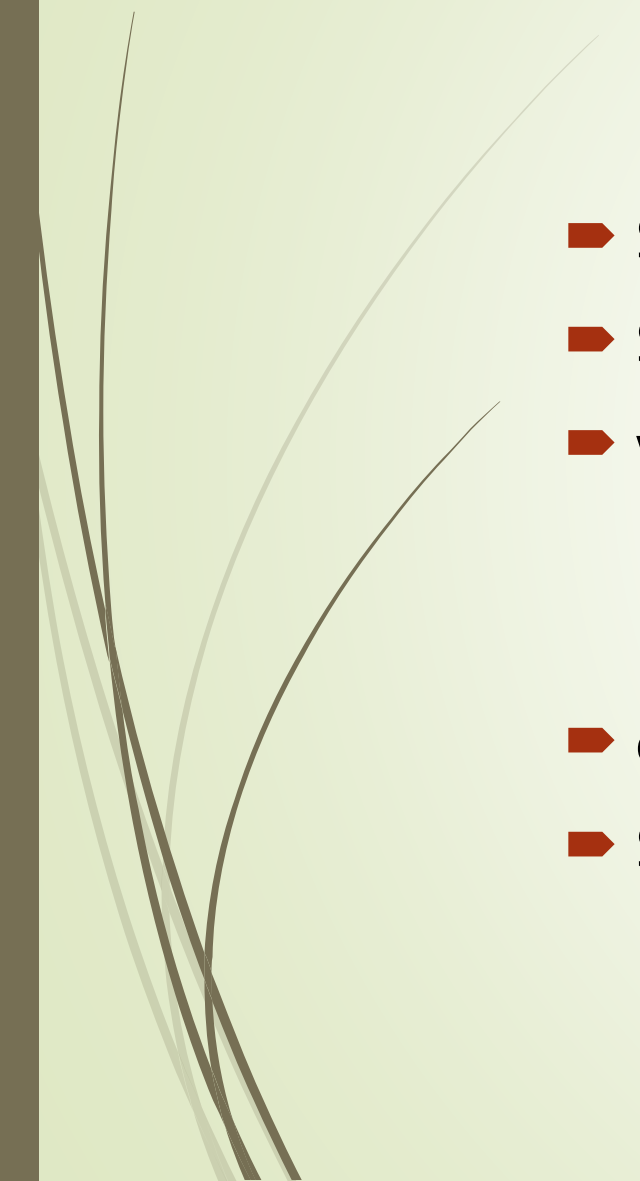


Print 1-1000 Natural Numbers

- Start
- Set counter = 1
- Set number = 1
- while(counter <= 1000)
 - Output number
 - increment number
 - increment counter
- end while
- Stop



Print 1-1000 Natural Numbers

- Start
 - Set counter = 1
 - while(counter <= 1000)
 - Output counter
 - increment counter
 - end while
 - Stop
- 



Exercise 1

Count = 1

While (Count < 5)

 output Count

 Count = Count + 1

End While





Exercise 2

Count = 1

While (Count < 5)

 Output " * "

 Count = Count + 1

End While



Exercise 3

Count = 1

```
While (Count < 5)
  output Count
  Count = Count + 1
End While
```

Count = 1

```
While ( Count < 5)
  Output " * "
  Count = Count + 1
End While
```


Exercise 4

Count = 1

Z = 1

While (Count < 10)

If Count == 3

 Z = 2

End if

Output Count " "

Output Z

Output Newline

Count = Count + 2

End While

Exercise 5

Input Variable: X

Count = 1

Z = 1

While (Count < 3)

Input X

If X is greater than 5

Z = 2

Else

Z = 4

End if

Count = Count + 1

End While

Output Z

What is output of above program
if input is

10 5

Exercise 6

Sum = 0

Count = 1

While (Count < 5)

Sum = Sum + Count

Count = Count + 1

End While

Output Sum



Exercise 7

Count = 1

While (Count > 0)
 Output Count
 Count = Count + 1
End While



Exercise 8

Count = 0

While (Count != 0)
 Output Count
 Count = Count + 1
End While

Exercise 9

Count = 1

While (Count < 5)

Output " * "

Count = Count + 1

If (Count == 4)

Output New line

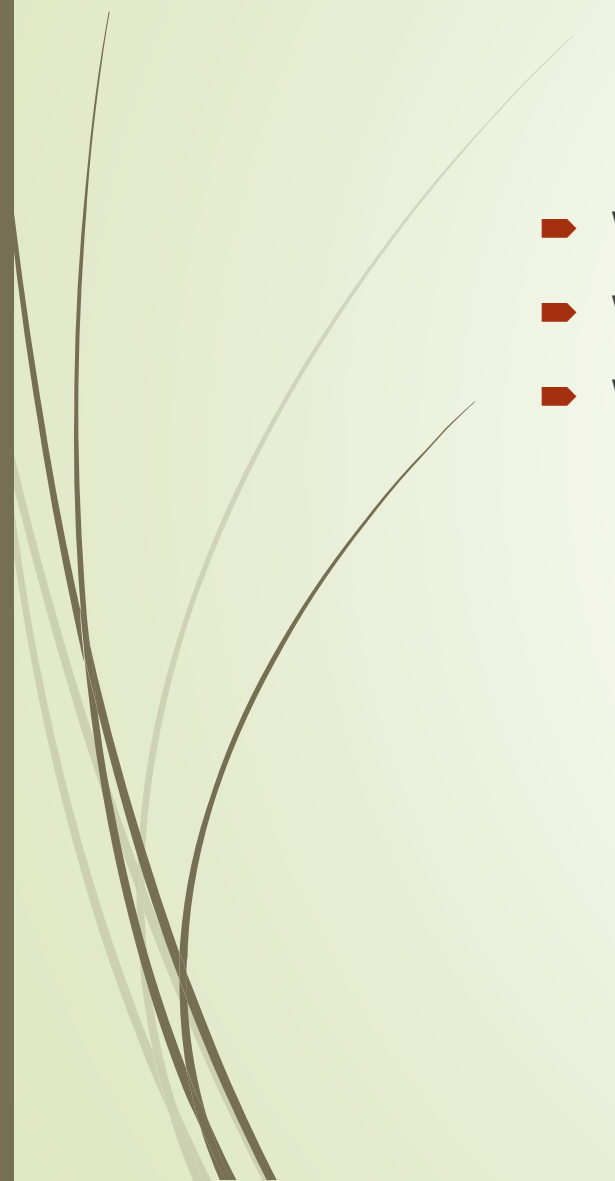
Count = 1


End if

End While



Practice

- Write code to calculate factorial of a number
 - Write code to calculate power of any number
 - Write code to print table of any number till 10
- 



Set Number1 = 1

Set Number2 = 2

Set Number3 = 3

Set value of Sum = Number1+Number2+Number3

while (sum <= 39)

begin while loop

 Number1 = Number2

 Number2 = Number3

 Number3 = Sum

 Sum = Number1+Number2+Number3

 output "*"

end of while loop

output "Number1 is :", output Number1

output "Number2 is:", output Number2

output "Number3 is:", output Number3

output "Sum is:", output Sum



START

Number1 = 1

Number2 = 30

Sum = 0

while (Number2 >= Number1)

begin while loop

 Number1 = Number1 + 1

 Number2 = Number2 - 2

 Sum = Number1 + Number2

end while loop

output "Number1 is :", output Number1

output "Number2 is:", output Number2

output "Sum is", output Sum

END



Exercise

- Write pseudocode for a program which reads 10 integers from the user, and output the largest value input.



Exercise

- Write pseudocode for a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".



References



1. C++ Programming: From Problem Analysis to Program Design, Third Edition
2. <https://www.just.edu.jo/~yahya-t/cs115/>