



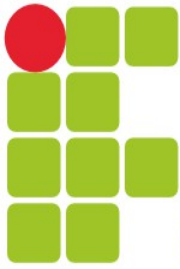
INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE  
Campus Parnamirim

# Preparatório OBI

Prof. André Gustavo Duarte de Almeida  
[andre.almeida@ifrn.edu.br](mailto:andre.almeida@ifrn.edu.br)  
[docente.ifrn.edu.br/andrealmeida](http://docente.ifrn.edu.br/andrealmeida)

Aula 02 – Estruturas de Controle e Decisão





INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE  
Campus Parnamirim

# Modelo Básico de Código C++

```
#include <iostream>
//importar bibliotecas
using namespace std;
//declarar funções ou tipos de dados
int main(){
    //código do método principal
    return 0;
}
```

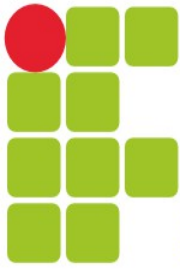
Arquivos com extensão .cpp

Para compilar um arquivo escrito em c++

Entre no diretório via terminal

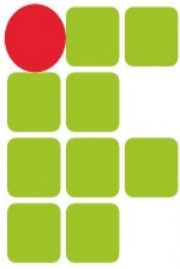
```
g++ -o <programa> <programa>.cpp
```





# Lendo Dados em C++

- CIN
- Realizar a leitura de dados, de qualquer tipo
- Exemplo:
- `int x,y;`
- `cin>>x>>y;`//Lê os valores de x e y digitados pelo usuário
- O operador `>>`, indica o fluxo, que é da entrada para o programa



# Exibindo Dados em C++

- COUT
- Mostra/imprime os dados na saída especificada
- Exemplo:
- `cout<<x<<"Teste<<endl;`
- O operador << indica o fluxo, que é de entrada para saída

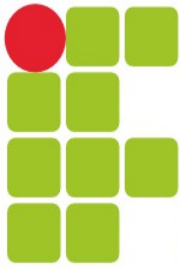


# Estrutura de Controles

- Comando IF
- Determinar que se uma condição for verdadeira um bloco de comandos deve ser executado

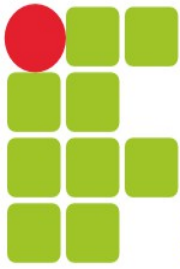
- Sintaxe 

```
If (<expressao_booleana>){  
//codigo  
}else{  
}  
if(<expressão_booleana>){  
}else if(<expressao_booleana>){  
}  
.....
```



# Estrutura de Controle

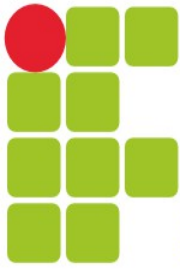
- Enquanto – while
- Repete um determinado bloco de comandos enquanto uma determina expressão booleana seja verdadeira
- Deve ser definido em 3 partes
- Inicialização
- Expressão de Avaliação
- Expressão para garantir o fim do loop



# Estrutura de Controle

- Enquanto - while

```
#include <iostream>
using namespace std;
int main(){
    int contadorNumerico=1;
    while(contadorNumerico<=10){
        cout<<(contadorNumerico*2)<<endl;
        contadorNumerico++;
    }
}
```



# Estrutura de Controle

- Do-While
- Semelhante ao while, porém a expressão de avaliação é analisada apenas no final do bloco, ou seja, pelo menos uma vez o bloco de repetição é executado

```
do
{
    cout << "Selecione a Opção " << endl;
    cout << "1) Adição" << endl;
    cout << "2) Subtração" << endl;
    cout << "3) Multiplicação" << endl;
    cout << "4) Divisão" << endl;
    cin >> nSelection;
} while (nSelection != 1 && nSelection != 2 &&
        nSelection != 3 && nSelection != 4);
```

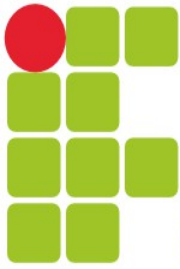




# Estrutura de Controle

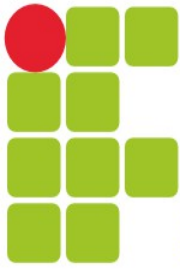
- Para – for
- Estrutura de repetição que define e uma única instrução as condições de inicialização, avaliação e de encerramento do loop

```
int nBase, int nExp;  
cin>>nBase>>nExp;  
int nValue = 1;  
for (int contador=0; contador < nExp; contador++){  
    nValue *= nBase;  
}  
cout<<nValue;
```



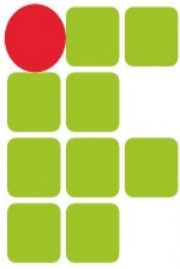
# Resolvendo o Problema

- Estratégia de Solução
- Ler o problema
- Observar os limites de entrada e características da saída
- Identificar qual(is) estrutura(s) de dados e controle são necessária
- Projetar inicialmente no papel, pensando em situações diferentes dos exemplos
- Codificar a solução



# Resolvendo o Problema

- Estratégias de Solução
  - Observe com atenção a natureza do problema
  - Problemas que envolve números podem necessitar atenção especial, em especial aqueles que envolvem número inteiros e reais devido a precisão
  - A estrutura de dado (vetor, matriz, variáveis comuns) a serem utilizadas são definitivas para a solução do problema
  - Em determinadas situações é melhor construir funções adicionais para modularizar a solução dos problemas



# Resolvendo o Problema

- Estratégias de Solução
  - Siga estritamente as orientações no que diz respeito a entrada e saída
  - Não use saídas de orientação ao usuário, como: “Digite o valor”/”Informe o nome”, apenas se for dito pelo problema
  - A correção do programa é feita pela saída. Uma letra errada ou uma quebra de linha indevida a solução é considerada errada

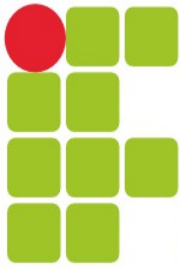


INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE  
Campus Parnamirim

# Resolvendo o Problema

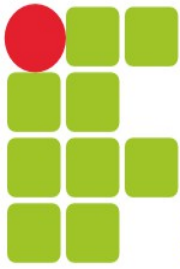
- Problema Inicial
- Nível: Fácil
- Aviões de Papel – Nível 01 – Fase 01 – Ano: 2009
- Link:  
[http://olimpiada.ic.unicamp.br/pratique/programacao/nivel1/2009f1p1\\_papel](http://olimpiada.ic.unicamp.br/pratique/programacao/nivel1/2009f1p1_papel)





# Resolvendo o Problema

- Analisando o problema
- “Você deve escrever um programa que, dados o número de competidores, o número de folhas de papel especial compradas pela Diretora e o número de folhas que cada competidor deve receber, determine se o número de folhas comprado pela Diretora é suficiente.”
- Observar o que é entrada:
  - Número de competidores
  - Número de folhas de papel especial
  - Número de folhas por competidor
- Saída
  - Sim → S ou Não → N



# Resolvendo o Problema

- Analisando o problema

- Questão, quando sabemos que o número de papéis é insuficiente???
- Múltiplas possibilidades
- Número Folhas Compradas / Número de Folhas Por participante  $<$  Número de Participantes
- Número de Participantes  $\times$  Número de Folhas por Participante  $>$  Número de Folhas Compradas
- **Tanto faz qualquer uma das soluções?!**



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE  
Campus Parnamirim

# Implementando

- Vamos a Implementação







# Treinando

## •Exercício

- Conta de Água
- Nível: Fácil
- <http://olimpiada.ic.unicamp.br/pratique/programacao/nivel1/2011>
- 

## •Exercício 02 – Usando Char

- Auto Estrada
- Nível: Fácil + Estrutura de Repetição
- <http://olimpiada.ic.unicamp.br/pratique/programacao/nivel1/2011>
-