Recap

- passaggio per valore
- passaggio per riferimento
- puntatori come parametri di funzioni
- valori di ritorno di funzioni
- metodi const

Valori di ritorno

- I valori di ritorno hanno le stesse regole dei parametri delle funzioni, possono essere tornati per valore, per riferimento, per puntatore

  int Add(int a, int b) //per valore

  int& Add(int a, int b) //per riferimento

  int* Add(int a, int b) //per puntatore

-

- Quali di queste funzioni è corretta? quale non lo è?

Passaggio per valore

- void Foo(int a)
- void Foo(MyClass A)
- void Foo(const MyClass A)

  a che serve const nell'ultimo esempio?

  void Main()
- {
- MyClass B;
- Foo(B);
- }
-

Passaggio per valore

- Supponiamo sizeof(B) = 10Byte e &B = 0xABC
- void Main()
- {
- MyClass B; //allocati 10 Byte
- Foo(B); //allocati 10 Byte per la copia di B
- }

  La funzione Foo è chiamata con una copia temporanea di B

  Foo(B') sizeof(B') = 10 Byte e &B' = 0xEDF
- sono due instanze diverse

Passaggio per valore

- void Add(int a, int b, int result) ?

  E' corretta?

passaggio per riferimento

- void Foo (int &a)
- void Foo (MyClass& a)
- void Foo (const MyClass&a)

  a che serve l'ultimo const?

  void Main()
- {
- MyClass B;
- Foo(B);
- }

passaggio per riferimento

- Supponiamo sizeof(B) = 10Byte e &B = 0xABC
- void Main()
- {
- MyClass B; //allocati 10 Byte
- Foo(B); //allocati 4 Byte per la copia dell'indirizzo
- }

  La funzione Foo è chiamata passando l'indirizzo di B

  Foo(B') sizeof(B') = 10 Byte e &B' = 0xABC
- è la stessa instanza

passaggio per riferimento

- void Add(int&a, int &b, int&c)

  è corretta?

puntatori come parametri di funzioni

- void Foo(int* a)
- void Foo(MyClass* A)
- void Foo(const MyClass* A)

  a che serve const nell'ultimo esempio?
- void Foo(const MyClass* const A)

  a che servono i due const nell'ultimo esempio

  void Main()
- {
- MyClass* B = new MyClass();
- Foo(B);
- }

puntatori come parametri di funzioni

- void Add(int* a, int* b, int* c)

  è corretta?

puntatori come parametri di funzioni

- sizeof(B) = 4 Byte e &B = 0xABC B=0xDFG sizeof(*B) = 10B
- void Main()
- {
- MyClass * B = new MyClass(); //allocati 4 Byte sullo stack e 10 sull'heap
- Foo(B); //allocati 4 Byte per la copia del puntatore
- }

  La funzione Foo è chiamata passando una copia del puntatore a B
- è la stessa instanza

metodi const

- void Foo(const int &a) const;
- int Add (const int a, const int b) const;
- const int Add(const int a, const int b) const;
- const int& Add(const int a, const int b) const;
- const int * Add(const int* a, const * b) const;

quali di questi è corretti?

RIferimento di un puntatore

- void Foo(int * a)
- void Foo(int &a)

- void OffsetInMemory(unsigned char * pInit, int offset)