



תכנות מונחה עצמים

מעבדה 2

SCE

המכללה האקדמית להנדסה סמי שמעון

תתחבר לרעיונות גדולים

באר שבע | אשדוד ■ www.sce.ac.il *מהנדס

העמסת פונקציות (Functions Overloading)

- שתי פונקציות או יותר בעלות שם זהה, אבל ארגומנטים (שונים) ידועות כפונקציות עמוסות.
- בשפת C++, שתי פונקציות יכולות להיות עם שם זהה אם מספר הארגומנטים / או סוג הארגומנטים שמועברים שונים.
- בשפת C++ החתימה של הפונקציה היא השם של הפונקציה ורשימת הארגומנטים שלה (לעומת שפת C בה החתימה של הפונקציה היא השם שלה בלבד).
- לא ניתן לבצע העמסת פונקציות אם הן נבדלות רק בערך המוחזר מהפונקציה - הערך המוחזר מהפונקציה אינו חלק מחתימת הפונקציה.

העמסת פונקציות (Functions Overloading) - דוגמא

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

void print(int i);
void print(double f);
void print(char* c);

void print(int i) {
    cout << "Printing int: " << i << endl;
}
void print(double f) {
    cout << "Printing float: " << f << endl;
}
void print(char* c) {
    cout << "Printing character: " << c << endl;
}

int main() {
    // Call print to print integer
    print(5);
    // Call print to print float
    print(500.263);
    // Call print to print character
    print("Hello C++");
    return 0;
}
```

העמסת פונקציות (Functions Overloading)

כאשר ישנה קריאה לפונקציה מועמסת C++ מחפש את הפונקציה המתאימה -

1. אם קיימת התאמה אחת או יותר תיקרא הפונקציה המתאימה ביותר – אם קיימת התאמה מדוייקת הקומפיילר יקרא לה, אחרת הקומפיילר יבצע implicit conversion.
2. כאשר אין התאמה או שיש יותר מפונקציה אחת המתאימה באותה קרבה נקבל שגיאת קומפילציה.

ambiguity (דו משמעות)

- כל ה-implicit conversion נחשבות שוות, וכל ההמרות המוגדרות על ידי משתמשים נחשבות שוות.
- לכן, אם קריאה לפונקציה מתאימה למספר פונקציות באמצעות implicit conversion או המרה המוגדרת על ידי משתמש, התוצאה תהיה ambiguity.

ambiguity (דו משמעות)

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

void print(float f);
void print(double d);
void print(char* c);

void print(double d) {
    cout << "Printing dpuble: " << d << endl;
}
void print(float f) {
    cout << "Printing float: " << f << endl;
}
void print(char* c) {
    cout << "Printing character: " << c << endl;
}

int main(void) {
    print(5);
    return 0;
}
```

more than one instance of overloaded function "print" matches the argument list:

⊗ C2668 'print': ambiguous call to overloaded function

Default Values (ערכי ברירת מחדל)

- ערכי ברירת מחדל לפרמטרים = האפשרות לתת לפרמטרים ערך דיפולטיבי במידה והמשתמש לא מכניס ערך משלו.
- בשפת C++ ניתן להגדיר ערכי ברירת מחדל לפרמטרים של פונקציה.
- התוצאה של מתן ערכי ברירת מחדל היא העמסת פונקציות.
- הפרמטרים המקבלים ערך ברירת מחדל חייבים להופיע בסוף רשימת הפרמטרים (יכול להיות שכל הפרמטרים יקבלו ערכי ברירת מחדל).
- במילים אחרות: אם נתנו ערך ברירת מחדל לפרמטר כלשהו, אנחנו חייבים לתת ערך ברירת מחדל לכל הפרמטרים המופיעים אחריו (מימינו).
- הדבר מאפשר להשמיט את הפרמטר בזמן הקריאה לפונקציה.
- אם משמיטים פרמטר אחד – יש להשמיט את כל הפרמטרים שאחריו (מימינו).