



# PROBLEM BASED APPROACH IN TEACHING OOP COURSES

ANA MADEVSKA BOGDANOVA

FCSE – UKIM

MACEDONIA

# OUTLINE

- WHAT IS PROBLEM BASED LEARNING
- WHY IS THIS APPROACH INTERESTING IN TEACHING OOP PROGRAMMING
- STUDENT – TEACHER ROLES
- PB ELEMENTS IN LECTURES
- CONCLUSION

# PBL? LEARNING BY POSING PROBLEMS, NOT SOLUTIONS

- PROBLEM-BASED LEARNING (PBL)
  - STUDENT CENTERED APPROACHES
- PBL IS A PEDAGOGICAL APPROACH HELPING STUDENTS DEVELOP
  - SELF- DIRECTED LEARNING SKILLS
  - CREATIVE THINKING,
  - PROBLEM SOLVING
  - COMMUNICATION SKILLS
- PBL – LEARNING AS A PROCESS IN WHICH THE LEARNER ACTIVELY CONSTRUCTS NEW KNOWLEDGE ON THE BASIS OF CURRENT KNOWLEDGE.

# WHY IS PBL IMPORTANT?

- THROUGH PROBLEM-BASED LEARNING (PBL), STUDENTS USE “TRIGGERS” FROM THE PROBLEM CASE OR SCENARIO TO DEFINE THEIR OWN LEARNING OBJECTIVES
- PBL IS NOT ONLY ABOUT PROBLEM SOLVING, BUT RATHER IT USES APPROPRIATE PROBLEMS TO INCREASE KNOWLEDGE AND UNDERSTANDING
  - PROBLEMS SHOULD BE ADEQUATE FOR THE CERTAIN LESSON
  - PROBLEMS SHOULD HAVE DISTINCTIVE ‘WEIGHT’

# STUDENT – TEACHER ROLES

Student

- Group works
- Practice
- Discussion
- Individual and peer

Teacher

- Guide
- Feedback
- Support

# FOUNDATION PROGRAMMING COURSES – FACTS

- THE FOUNDATION COURSES IN COMPUTER SCIENCE DEVELOP
  - BASIC SKILLS
  - ATTITUDES WHICH ARE IMPORTANT FOR EFFECTIVE LEARNING IN LATER COURSES
- THE COURSES ARE TYPICALLY BASED UPON INDIVIDUAL WORK. BY CONTRAST, MUCH OF THE PROGRAMMING WORKFORCE OPERATES IN TEAMS.
- A LARGE PROPORTION OF COMPUTER SCIENCE GRADUATES WILL FIND EMPLOYMENT WHICH INVOLVES BROAD PROBLEM SOLVING SKILLS, RATHER THAN PURELY TECHNICALLY CENTERED ACTIVITY.
  - THE FOUNDATION COURSES USUALLY HAVE A NARROW TECHNICAL FOCUS
- THE MAJORITY OF STUDENTS IN THE FIRST YEAR COMPUTING COURSES ARE ALSO IN THEIR FIRST YEAR AT UNIVERSITY.
  - THESE STUDENTS ARE PARTICULARLY FRAGILE.



# POTENTIAL PROBLEMS

- IN A CONVENTIONAL COURSE, WE USE AUTOMATED GRADING.
  - MAKES THE GRADING MORE MANAGEABLE.
  - BUT, BY LATTER COURSES, STUDENTS EXPECTS PROGRAMMING TASKS TO HAVE EXTREMELY TIGHT SPECIFICATIONS AND THEY RELIED ON US TO SET THE STANDARDS WHICH DEFINED THE SUCCESS OF THEIR WORK.
- DIVERSITY IN OUR LARGE STUDENT BODY
  - SOME HAVE BEEN PROGRAMMING SINCE THEY WERE QUITE YOUNG.
  - MOST HAVE NO PROGRAMMING EXPERIENCE AT ALL.
- AN IMPORTANT DIMENSION OF THE OO PARADIGM COMES FROM THE WRITING OF LARGE PROGRAMS WHICH INVOLVE SEVERAL PROGRAMMERS.
  - IF EACH STUDENT DOES ASSIGNMENTS ALONE, IT IS MORE DIFFICULT FOR THEM TO APPRECIATE THIS ASPECT.

# FORMING THE STATE OF MIND

- THE NATURE OF PROGRAMMING REQUIRES A SPECIFIC TYPE OF THINKING THAT NOVICE STUDENTS ARE NOT FAMILIAR WITH.
- INSTEAD OF FOCUSING JUST ON THE INTENDED EXECUTION PATH OF THE PROGRAM, THE STUDENT MUST ALSO CONSIDER ALL THE UNINTENDED OR EXCEPTIONAL SITUATIONS THAT MAY BE ENCOUNTERED IN THE EXECUTION



# HOW TO USE PBL IF...

- MOST COMPUTING COURSES INVOLVE SETTING 'PROBLEMS' WHICH STUDENTS ARE REQUIRED TO COMPLETE.
  - ACTUALLY EXERCISES, BECAUSE THEY ARE SMALL AND WELL-DEFINED
- WEEKLY EXERCISES, EACH FOCUSED ON PARTICULAR DETAILED ASPECTS OF THE COURSE, USUALLY ONE THAT HAD BEEN ON CENTER STAGE IN THE RECENT LECTURES;
- LARGER ASSIGNMENTS WHICH INTEGRATES MANY ASPECTS OF THE COURSE BUT WERE STILL QUITE TIGHTLY DEFINED, IN ORDER TO BE EVALUATED IN OUR AUTOMATIC GRADING SYSTEM.

# SO, HOW TO USE PBL?

- *STARTING POINT FOR LEARNING SHOULD BE A PROBLEM, A QUERY OR A PUZZLE*
- *USE STIMULUS MATERIAL TO ENGAGE STUDENTS IN CONSIDERING A PROBLEM WHICH IS PRESENTED IN THE SAME CONTEXT AS THEY WOULD FIND IT IN “REAL LIFE”*
- *INFORMATION ON HOW TO TACKLE THE PROBLEM IS NOT GIVEN,*
  - *RESOURCES ARE AVAILABLE TO ASSIST THE STUDENTS TO CLARIFY WHAT THE ‘PROBLEM’ CONSISTS OF AND HOW THEY MIGHT DEAL WITH IT.*
- *STUDENTS SOMETIMES WORK COOPERATIVELY IN A GROUP OR TEAM*
- **WORK WITH MUCH BROADER PROBLEMS WHICH INVOLVE A LARGER SET OF PROBLEM SOLVING SKILLS.**

# FEW STEPS OF USING PL APPROACH

Step 1: Examination of the case. The group gets familiar with the case material.

Step 2: Identification of the problem. An initial title for the case is specified.

Step 3: Brainstorming. The students present their associations and ideas about the problem to find out what is already known and how does the case relate to the previous knowledge. The ideas are said aloud and written on self-stick notes, which are organized on a white board.

Step 4: Sketching of an explanatory model. An initial version of the explanation for the problem is constructed and most important concepts and their relations are identified.

# THE METHOD IS DESIGNED TO FOSTER LEARNING BY

...

1. CONNECTING THE LEARNING TO *SPECIFIC PROBLEM SITUATIONS* THAT MAY BE ENCOUNTERED IN PRACTICE
2. *ACTIVATING PRIOR KNOWLEDGE* OF STUDENTS ABOUT THE TOPIC TO LEARN
3. MAKING THE STUDENTS TO *ELABORATE THE MATERIAL* THAT THEY HAVE LEARNED

# EXAMPLES

The image features a light blue gradient background. The word "EXAMPLES" is centered in a bold, black, sans-serif font. The corners of the image are decorated with several realistic water droplets of various sizes, each with a white highlight and a soft shadow, giving them a three-dimensional appearance.



# A TASK, THE OUTLINE (MULTIPLE INHERITANCE)

- A COMPANY FOR WHOSE EMPLOYEES IT HAS TO PAY A SALARY.
- THE COMPANY HAS DIFFERENT EMPLOYEE PROFILES, WHOSE SALARY IS CALCULATED DIFFERENTLY.
- SALARIES OF COLLABORATORS DO NOT DEPEND ON THE VALUE OF THE POINT AND ARE DETERMINED DIRECTLY.
- **EMPLOYEES:**
  - **DIRECTOR** – RECEIVES A FIXED FEE;
  - **MECHANIC** – PAID ACCORDING TO THE NUMBER OF WORKING HOURS WORKED IN THE MONTH, ACCORDING TO A CERTAIN FEE FOR 1 HOUR;
  - **WORKDEALER** – PAID ACCORDING TO THE NUMBER OF SALES ACHIEVED (GETS A CERTAIN BONUS FROM EACH SALE, BUT NO WORKING HOURS);
  - **VENDOR** – HAS A FIXED FIXED BASIS FOR THE SALARY TO WHICH INCENTIVE BONUSES ARE ADDED ACCORDING TO THE SALES MADE (THERE IS A SHOP FLOOR)
- **ASSOCIATES :**
  - **FREELANCE** – TEMPORARILY HIRED BY THE FIRM BY CONTRACT FOR A CONTRACT FOR AN AGREED AMOUNT THAT DOES NOT DEPEND ON THE VALUE OF THE POINT.
  - **CONSULTANT:** ENGAGED LUMP-SUM WITH A (MULTI) ANNUAL CONTRACT WITH A SPECIFIC PAYOUT DYNAMICS (MONTHLY, FOR EXAMPLE). UNDER THE AGREEMENT, IN SOME CASES, IT ALSO PAYS BONUSES FOR COSTS (ABOVE THE AGREED AMOUNT). IT DOES NOT DEPEND ON THE VALUE OF THE POINT.



# FIRST ITERATION (1)

```
#INCLUDE <Iostream>
#include <string>
using namespace std;
```

```
class RABOTNIK {
public:
    RABOTNIK(char *name = ""):IME(name) { } //
    KONSTRUKTOR

    virtual ~RABOTNIK() { }

    static void setBOD(float newBOD) { BOD=newBOD; }

    const string &IME(void) const { return IME; }

    float vrednostBOD(void) const { return BOD; }

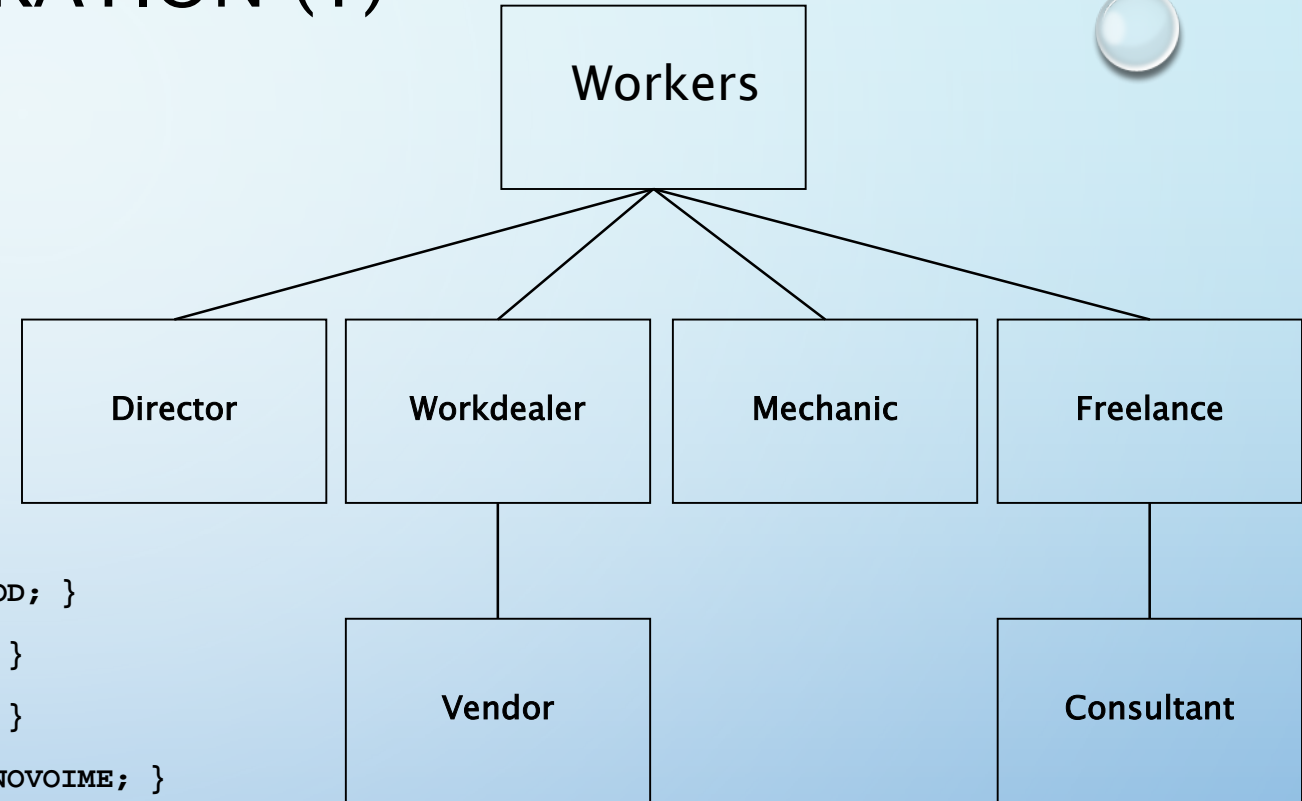
    void smeniIME(const string &noVOIME) { IME=noVOIME; }

    virtual float plata(void) const = 0;

    virtual void show(void) const = 0;

private:
    string IME;

    static float BOD;
};
```



# FIRST ITERATION (2)

```
CLASS DIREKTOR : PUBLIC RABOTNIK {
PUBLIC:
    DIREKTOR(CHAR *IME = "", FLOAT PLATA = 0): RABOTNIK(IME), BOD_PLATA(PLATA) {}
    ~DIREKTOR() { }
    FLOAT PLATA(VOID) CONST { RETURN BOD_PLATA*VREDNOSTBOD(); }
    VOID SHOW(VOID) CONST { COUT << IME() << ' ' << PLATA(); }
    VOID OSNOVICA(FLOAT VREDNOST) { BOD_PLATA = VREDNOST; }
PRIVATE:
    FLOAT BOD_PLATA;
};

CLASS MEHANICAR : PUBLIC RABOTNIK {
PUBLIC:
    MEHANICAR(CHAR *IME = "", FLOAT PLATA_PO_CAS = 0, INT RAB_CASOVI = 0):
        RABOTNIK(IME), SATNINA(PLATA_PO_CAS), CASOVI(RAB_CASOVI) { }
    ~MEHANICAR() { }
    FLOAT PLATA(VOID) CONST { RETURN SATNINA*CASOVI*VREDNOSTBOD(); }
    VOID SHOW(VOID) CONST { COUT << IME() << ' ' << PLATA(); }
    VOID RABOTNICASOVI(INT CAS) { CASOVI = CAS; }
    VOID SATNICA(FLOAT VREDNOST) { SATNINA = VREDNOST;}
PRIVATE:
    FLOAT SATNINA;
    INT CASOVI;
};
```

# FIRST ITERATION (3)

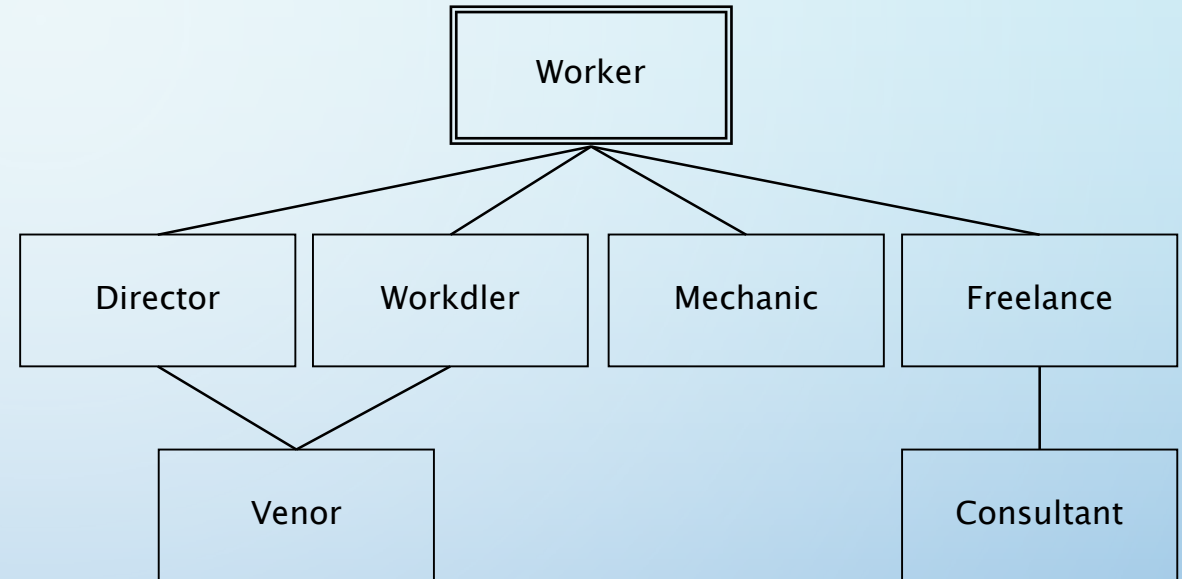
```
CLASS DILER : PUBLIC RABOTNIK
{
PUBLIC:
    DILER(CHAR *IME = "", FLOAT PROV = 0, INT BRPROD = 0):
        RABOTNIK(IME), BONUS(PROV), PRODAZBI(BRPROD) { }
    ~DILER() { }
    FLOAT PLATA(VOID) CONST { RETURN BONUS*PRODAZBI*VREDNOSTBOD(); }
    VOID SHOW(VOID) CONST { COUT << IME() << ' ' << PLATA(); }
    VOID BONUS(FLOAT VREDNOST) { BONUS = VREDNOST; }
    VOID PRODAZBI(INT BR) { PRODAZBI = BR; }
PRIVATE:
    FLOAT BONUS;
    INT PRODAZBI;
};

CLASS PRODAVAC : PUBLIC DILER
{
PUBLIC:
    PRODAVAC(CHAR *IME = "", FLOAT OSNOVA = 0, FLOAT PROV = 0, INT BRPROD = 0):
        DILER(IME,PROV,BRPROD), MESECNA(OSNOVA) { }
    ~PRODAVAC() { }
    FLOAT PLATA(VOID) CONST { RETURN MESECNA*VREDNOSTBOD()+DILER::PLATA(); }
    VOID SHOW(VOID) CONST { COUT << IME() << ' ' << PLATA(); }
    VOID OSNOVICA(FLOAT VREDNOST) { MESECNA = VREDNOST; }
PRIVATE:
    FLOAT MESECNA;
};
```

# SECOND ITERATION

```
#INCLUDE <Iostream>
#include <string>
using namespace std;
```

```
class Rabotnik {
public:
    . . .
};
class Direktor : virtual public Rabotnik {
    . . .
};
. . .
class Prodavac : public Direktor, public Diler
{
public:
    Prodavac(char *ime = "", float osnova = 0, float prov = 0, int brprod = 0)
    { smeniime(ime); bonus(prov); prodazbi(brprod); osnovica(osnova); }
    ~Prodavac() { }
    float plata(void) const { return Direktor::plata()+Diler::plata(); }
    void show(void) const { cout << ime() << ' ' << plata(); }
};
. . .
```



# THE FINAL VERSION

```
#INCLUDE <Iostream>
#include <string>
using namespace std;
```

```
class RABOTNIK {
public:
```

```
    RABOTNIK(char *name = ""):ime(name) { } // KONSTRUKTOR
```

```
    virtual ~RABOTNIK() { }
```

```
    const string &ime(void) const { return ime; }
```

```
    void smeniime(const string &novoime) { ime=novoime; }
```

```
    virtual float plata(void) const = 0;
```

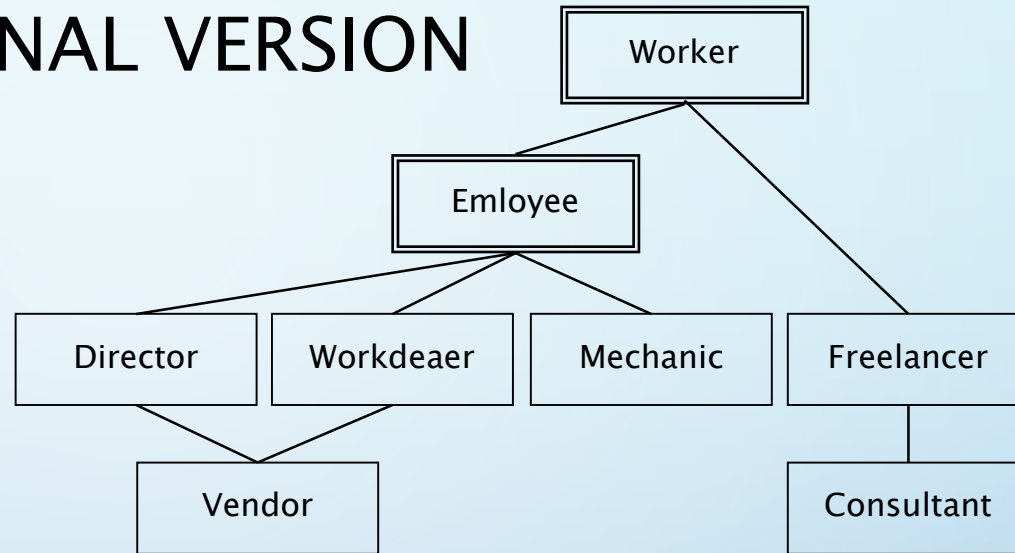
```
    void show(void) const
```

```
        { cout << ime() << " " << plata(); }
```

```
private:
```

```
    string ime;
```

```
};
```



## THE FINAL VERSION (2)

```
CLASS VRABOTEN: PUBLIC RABOTNIK
{
PUBLIC:
    VRABOTEN(CHAR *NAME = ""):
        RABOTNIK(NAME) { } // KONSTRUKTOR
    VIRTUAL ~VRABOTEN() { }
    STATIC VOID SETBOD(FLOAT NEWBOD) { BOD=NEWBOD; }
PROTECTED:
    STATIC FLOAT BOD;
};
```



# THE FINAL VERSION (3)

```
CLASS DIREKTOR : VIRTUAL PUBLIC VRABOTEN
```

```
{  
    PUBLIC:  
    DIREKTOR(CHAR *IME = "", FLOAT PLATA = 0): VRABOTEN(IME), BOD_PLATA(PLATA) {}  
    ~DIREKTOR() { }  
    FLOAT PLATA(VOID) CONST { RETURN BOD_PLATA*BOD; }  
    VOID OSNOVICA(FLOAT VREDNOST) { BOD_PLATA = VREDNOST; }  
    PRIVATE:  
    FLOAT BOD_PLATA;  
};
```

```
CLASS MEHANICAR : PUBLIC VRABOTEN
```

```
{  
    PUBLIC:  
    MEHANICAR(CHAR *IME = "", FLOAT PLATA_PO_CAS = 0, INT RAB_CASOVI = 0):  
        VRABOTEN(IME), SATNINA(PLATA_PO_CAS), CASOVI(RAB_CASOVI) { }  
    ~MEHANICAR() { }  
    FLOAT PLATA(VOID) CONST { RETURN SATNINA*CASOVI*BOD; }  
    VOID RABOTNICASOVI(INT CAS) { CASOVI = CAS; }  
    VOID SATNICA(FLOAT VREDNOST) { SATNINA = VREDNOST; }  
    PRIVATE:  
    FLOAT SATNINA;  
    INT CASOVI;  
};
```

# THE FINAL VERSION(4)

```
CLASS DILER : VIRTUAL PUBLIC VRABOTEN
{
PUBLIC:
    DILER(CHAR *IME = "", FLOAT PROV = 0, INT BRPROD = 0):
        VRABOTEN(IME), BONUS(PROV), PRODAZBI(BRPROD) { }
    ~DILER() { }
    FLOAT PLATA(VOID) CONST { RETURN BONUS*PRODAZBI*BOD; }
    VOID BONUS(FLOAT VREDNOST) { BONUS = VREDNOST; }
    VOID PRODAZBI(INT BR) { PRODAZBI = BR; }
PRIVATE:
    FLOAT BONUS;
    INT PRODAZBI;
};

CLASS PRODAVAC : PUBLIC DIREKTOR, PUBLIC DILER
{
PUBLIC:
    PRODAVAC(CHAR *IME = "", FLOAT OSNOVA = 0, FLOAT PROV = 0, INT BRPROD = 0)
    { SMENIIME(IME); BONUS(PROV); PRODAZBI(BRPROD); OSNOVICA(OSNOVA); }
    ~PRODAVAC() { }
    FLOAT PLATA(VOID) CONST { RETURN DIREKTOR::PLATA()+DILER::PLATA(); }
PRIVATE:
    FLOAT MESECNA;
};
```

# WE ASKED STUDENTS ABOUT THE PBL APPROACH

- THE STUDENTS REPORTED THAT GROUP WORK IN PBL:
  - IMPROVED THE MOTIVATION,
  - PROVIDED EMOTIONAL SUPPORT,
  - GAVE A SOCIAL CONTEXT FOR THE COURSE
  - IT'S FUN
- WE SHOULD MAKE THIS APPROACH SYSTEMATIC IN THE WHOLE COURSE AND MEASURE THE BENEFIT OF THIS LEARNING METHOD, COMPARED TO THE CURRENT ONE
  - THE DROP-OUT RATE
  - THE SUCCESS IN THE FOLLOWING PROGRAMMING COURSES

# AS A CONCLUSION...

- IN SUMMARY, PBL IS CHARACTERIZED BY:
  - OPEN-ENDED PROBLEMS WHICH DRIVE THE LEARNING;
  - EXPLICIT TEACHING AND IMPROVING OF GENERIC AND METACOGNITIVE SKILLS;
  - COLLABORATIVE LEARNING IN GROUPS
  - IT IS FUN FOR THE TEACHER AS WELL
- IT IS GOOD IDEA TO BE INCLUDED IN THE OOP CURSES, AT LEAST PARTIALLY

QUESTIONS?

