# ESc101 : Fundamental of Computing

I Semester 2008-09

Lecture 14+15

**Methods**

## A typical JAVA program

```
class anyname
{
  public static void main(String args[])
  {
    int i;
    .
    .
    statements.
    .
    .
  }
}
```

## Method

**Definition :** a block of code which performs some well defined computational task.

- a method may have some input
- a method may also have some output

Example :

```
void main(String args[])
```

## Syntax of a method

public static *return_type* **method_name**(*input*)

{

**Body**

}

if there is no input parameter, we leave the parenthesis blank.

## Procedure : method which do not return anything

```
public static void Print_a_star()
{
  System.out.println(‘‘*’’);
}
```

## A method can be executed/called/invoked from other methods

```
class print_star
{
  public static void Print_a_star()
  {
    System.out.print(‘‘*’’);
  }

  public static void main(String args[])
  {
    Print_a_star();
  }
}
```

We say :

In the above program **Print_a_star** method is called from **main** method.

## Procedure with inputs

```
class print_stars
{
  public static void Print_multiple_stars(int i)
  {
    int k=1;
    for(;k<=i;k=k+1)
        System.out.print(‘‘*’’);
  }
  public static void main(String args[])
  {
    int j=5;
    Print_multiple_stars(j);
    System.out.println(‘’’’);
  }
}
```

## Procedures with inputs

```
public static void Print_multiple_stars(int i)
{
  statements;
}
```

- the value passed to a method is called its **argument**.
- The variable which receives the value is called **parameter**.
- parameters are declared inside the parenthesis and we must declare type of the parameters.
- **argument** may be an expression but

`type`(**argument**)=`type`(**parameter**)

## Procedures with inputs

```
void Procedure_name (parameter_list)
{
statements;
}
```

**For example :**

```
void Print_sum_of_numbers(int i, int j)
{
System.out.println(i+j);
}
```

is a method with two parameters `i` and `j` of type int, and it prints their sum.

## Two tips for good programming practice

- Come up with that identifier for a method which reflects its goal, that is, what it is doing.
- Write a line of comment after/before the method to convey what the method is supposed to do.

## Why should we use methods ?

## Why should we use methods ?

Examples in the next slides will convey their importance

## Example 1 :

Write a program to print

*  j times

\+  j+1 times

\-  j+2 times

**all in a line**

j is variable of type int.

## Example 1 :

There are two ways

1. without using methods

2. using methods

**ATTENTION :**

The file **print_using_main_only.java** available on the course website is the program without methods.

The file **print_stars_pluses_minuses.java** is the program using methods.

Please go through them.

## Advantages of using method

1. Avoids repetition of a piece of code

2. Improves the readability

## Advantages of using method

1. Avoids repetition of a piece of code

2. Improves the readability

Are there other advantages ?

## Advantages of using method

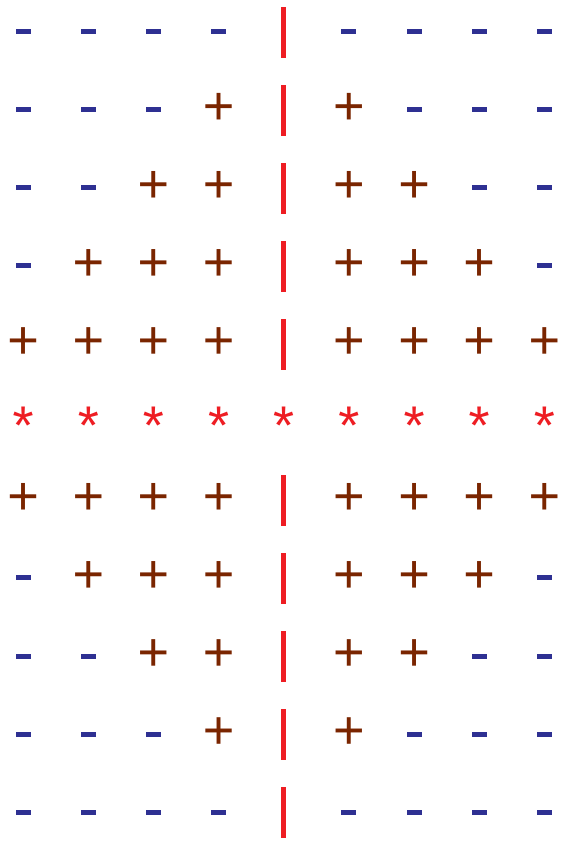
1. Avoids repetition of a piece of code

2. Improves the readability

The most important advantage will become clear from Example 2

## Example 2 : Printing a diamond $i$ times

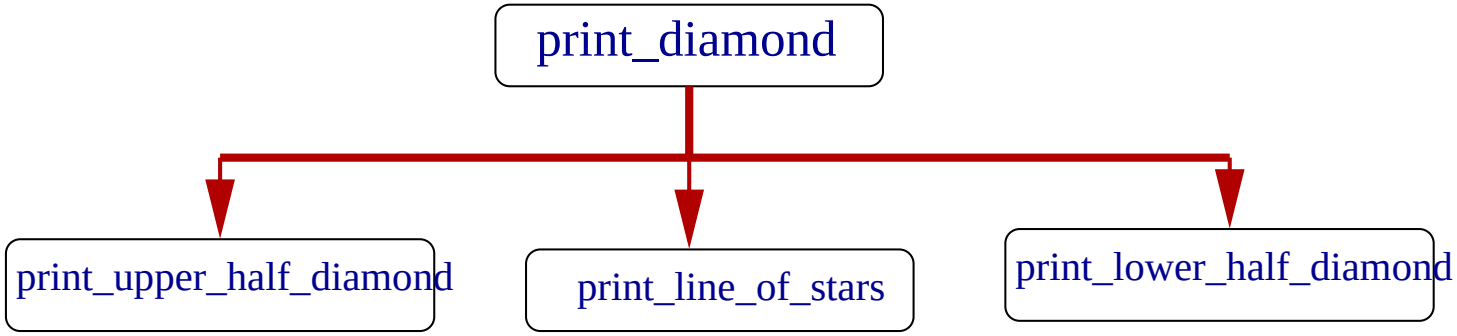
**half width** of diamond = $x$.



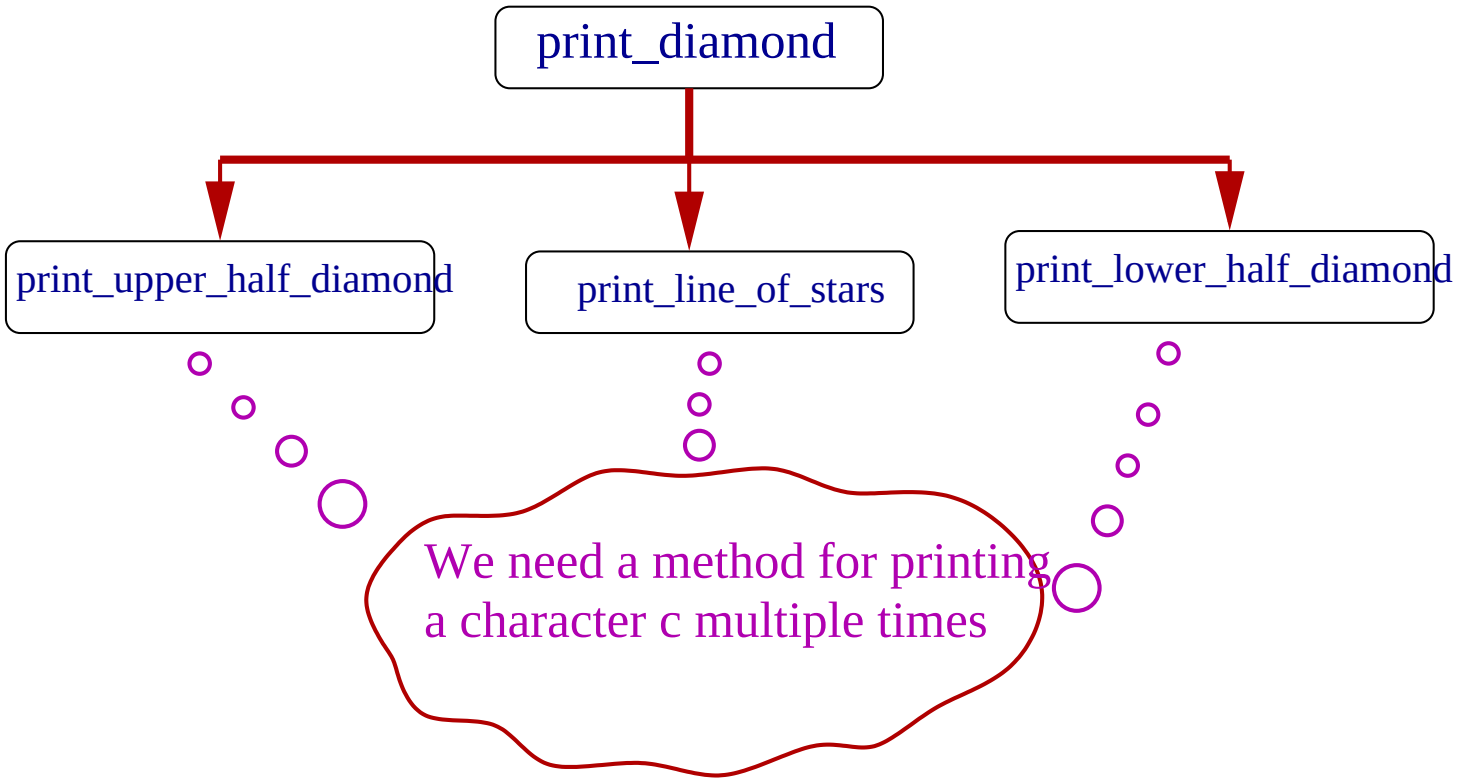
## printing a diamond

print_diamond

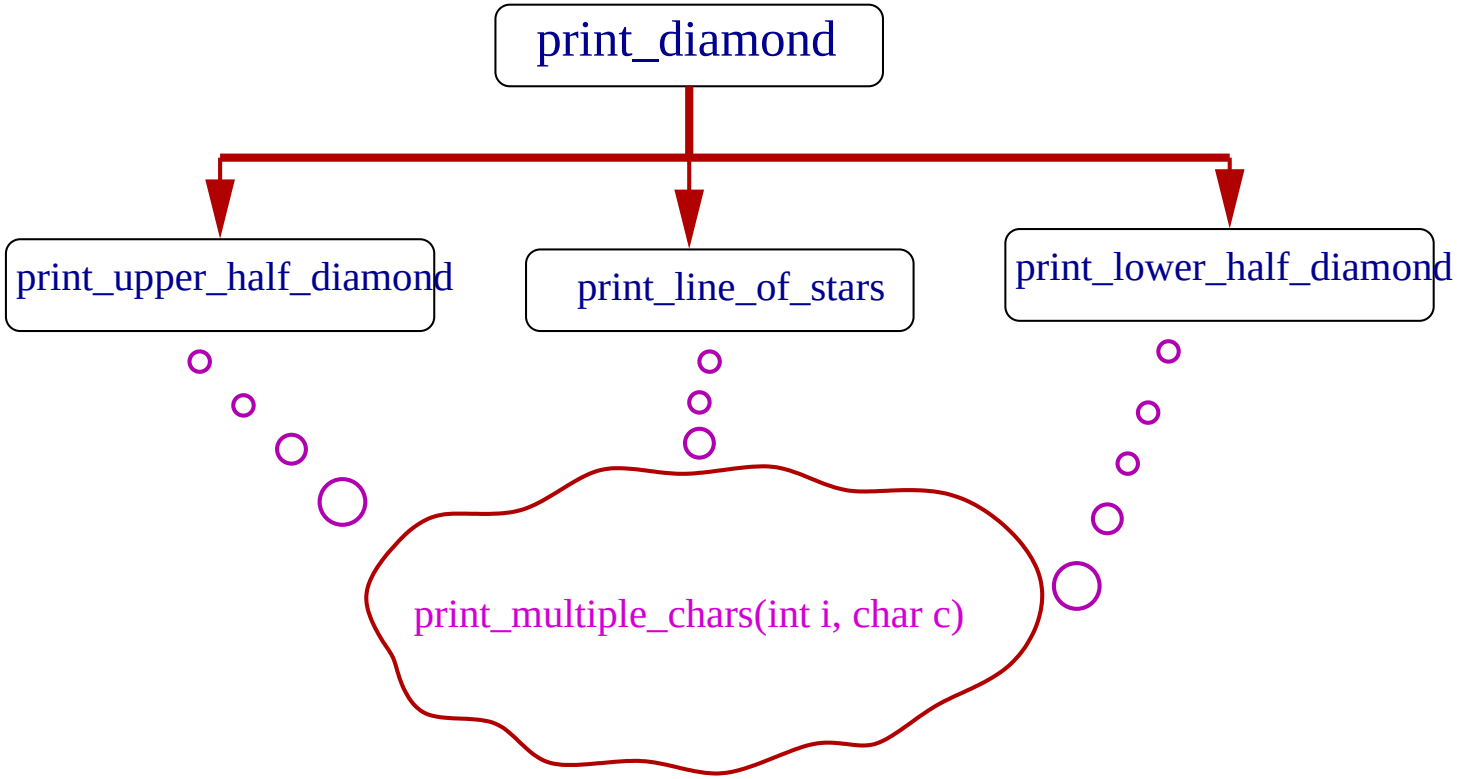
## printing a diamond



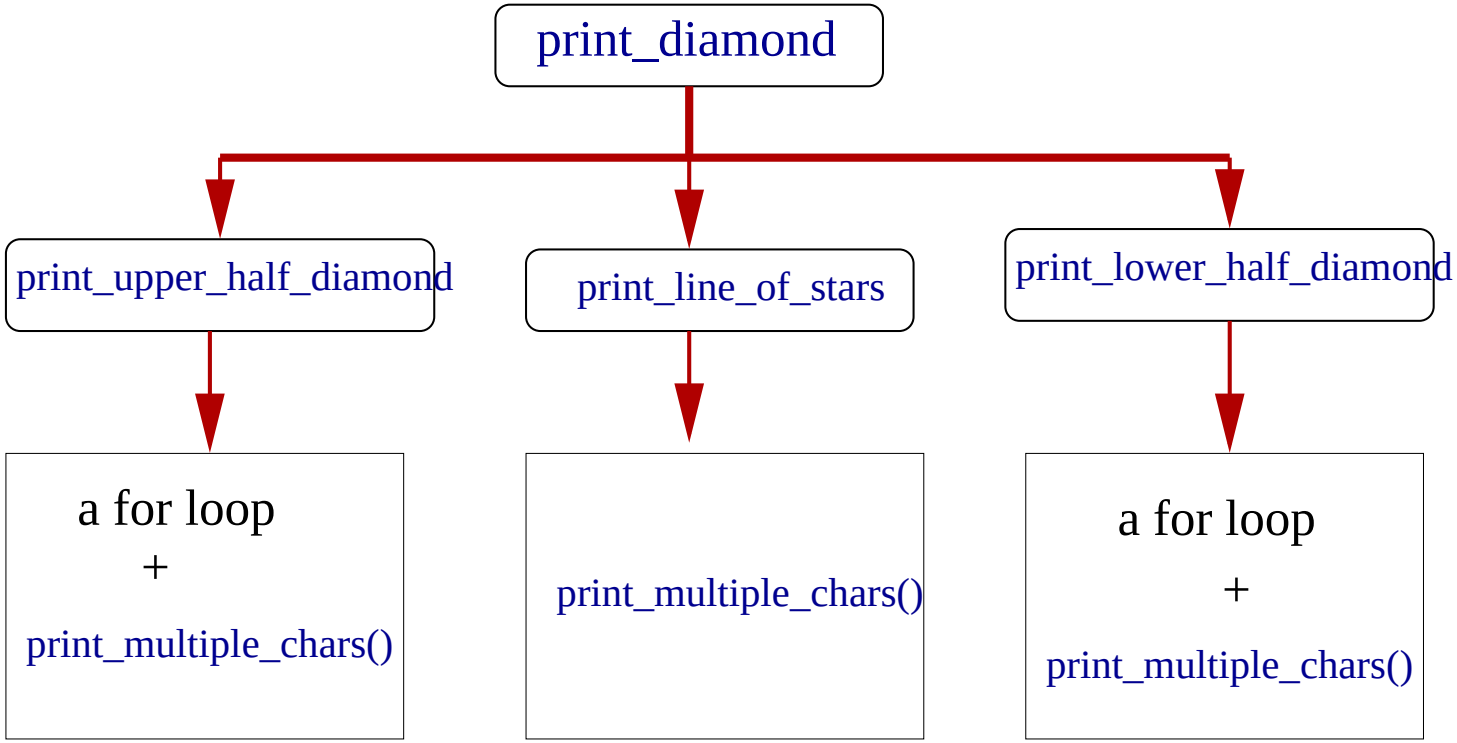
## printing a diamond



## printing a diamond



## printing a diamond



## Observation from Example 2

- To solve a problem, divide it into smaller subproblem.
- Divide each smaller sub-problem into further smaller sub-sub-problems until they become easy to solve (code should be 10-20 lines).
- finally for each problem, sub-problems,sub-sub-problems, write a separate method.

**A structured (top-down) way of writing the program**

## printing a diamond

- the file **bad_diamond.java** is the program (available on the course website) which does not use methods.
- the file **realdiamond.java** is the program (also available on the course website) which is using methods based on the top-down way discussed in previous five slides.

It is easy to see that the code of **realdiamond.java** is much more structured, elegant, and readable. Whereas the code of **bad_diamond.java** is more prone to logical errors since there are three nested for loops.

## An important advantage of using methods

**a structured way to write program**

which is good because

- It is `an easier way to design solution` of the problem.
- It is `easy to code` since each method will be a small piece of code.
- The program becomes `more manageable`.
- It `reduces the chances of logical errors`.

There are few more advantages to be discussed in future classes.