ESc101 : Fundamental of Computing

I Semester 2008-09

Lecture 16

Methods continued ...



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

Methods offer a structured (top-down) way of writing the programs

- 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-problems, write a separate method.

Advantages of structured programming

- 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.

Two kinds of methods

- Procedures methods which do not return anything
- Functions methods which do return something.

We discussed Procedures in the last two lectures

We shall discuss Functions in this lecture

An example of function

```
public static boolean Is_even(int k)
{
    if (k%2==0) return true;
    else return false;
}
We need to discuss two things:
    How do we call a function ?
    How do we execute a function ?
```

How do we call a function ?

type_t method_name(parameter_list)

- from any method
- Using statement :

variable_v = method_name(argument_list);

where type of variable_v must be $\geq type_t$.

Steps in execution of a function

Let function be : type_t method_name(parameter_list) and it is called by some method as :

```
var = method_name(argument_list);
```

Following steps are followed.

- 1. arguments are evaluated.
- 2. the parameters are initialized with the values of arguments.
- 3. the body of the function is executed
- 4. the function terminates on reaching statement

return expression

5. *expression* is evaluated and its value is returned to the calling method and assigned to **var**.

A complete example including calling the function

```
class check_even
{
   public static boolean Is_even(int k)
   {
      if (k%2==0) return true;
      else return false;
   }
   public static void main(String args[])
   {
      boolean flag; int i=11; int j=21;
      flag = Is_even(i*j);
      System.out.println(``IS i*j EVEN : ''+flag);
}
```

What about visibility of variables between two methods ?



When method M1 calls method M2

Let M1 calls M2. The following facts should be noted.

- M1 passes only the value of arguments to M2.
- M2 returns only a value computed to M1.

Consequences :

variables of **M1** are not visible inside **M2**, and variables of **M2** are not visible inside **M1**.

A question based on the facts of previous slide

```
class strange
{ public static int FUNC1(int i)
  { i = i+10;
    System.out.println(i);
    return 2*i;
  public static void main(string Args[])
  { int i = 1;
    int j = FUNC1(i);
    System.out.println(i+'',''+j);
  } }
```

The output will be : ??

A question based on the facts of previous slide

```
class strange
{ public static int FUNC1(int i)
  { i = i+10;
    System.out.println(i);
    return 2*i;
  }
  public static void main(string Args[])
  { int i = 1;
    int j = FUNC1(i);
    System.out.println(i+'',''+j);
  } }
```

The output will be :

11

1,22

On compatibility of types of argument and parameter

Let function be : $type_t$ method_name($type_a$ a)

and it is called by some method as :

var = method_name(expression);

Since *expression* is evaluated and its value is assigned to parameter,

\Rightarrow

We follow same rule for types as in case of the assignment statement.

type_a a = expression

Problems to be solved by structured programming

Design a JAVA program to solve the following problem :

Let i be an integer. Max(i) is the integer which is largest number formed by permuting the digits of i. For example, Max(1294) = 9421.

Design a JAVA program to compute Max(i) for a given integer i.

NOTE : Please follow the structured programming approach (similar to printing a diamond). Put sincere efforts to solve this problem befoe coming to lecture on Monday.