FACULTY OF ENGINEERING & TECHNOLOGY

Brajesh Mishra
Assistant Professor
Department of Computer Science & Engineering

Topics Covered

Stack-Based Storage Allocation
Heap-Based Storage Allocation
Sequence Control

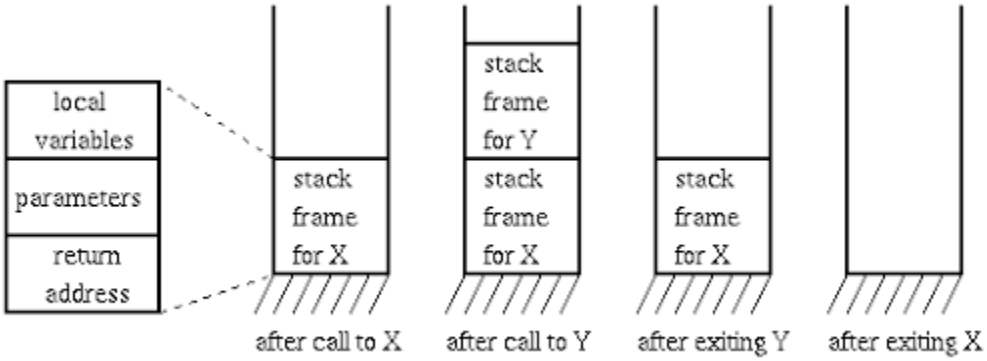
Stack-Based Storage Allocation

- Stack-based storage allocation is appropriate when the storage requirements are not known at compile time, but the requests obey a last-in, first-out discipline.
- Examples:

- local variables in a procedure in C/C++, Ada, Algol, or Pascal

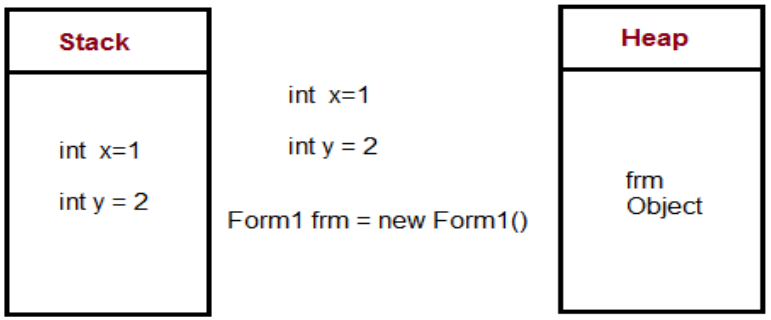
- procedure call information (return address etc).

```
import java.awt.Point;
class Squid
{ public static void main(String[] args)
{ int n = 1;
Point p = new Point(10,20);
Point q;
q = test(n,p); }
public Point test(int i, Point r)
{ Point s;
 s = new Point(r.x+i, r.y+i);
return s; } }
```



Heap-Based Storage Allocation

- The most flexible allocation scheme is heap-based allocation.
- Allocation is easy.
- In C, malloc (a standard library function) allocates fresh storage.
- In Lisp/Scheme, a new cons cell is allocated when the cons function is called, array storage can be allocated using make-array, and so forth.
- In Java new storage is allocated when the program makes a new instance of a class.
- Deallocation is harder.
- There are two approaches:
  - programmer-controlled
  - automatic.



Sequence Control

- The control of execution of the operations, both primitive and user defined, is termed as sequence control
- Sequence control structures are categorized into following four groups:

- **Expressions:**
  - These form the basic building blocks for statements and express how at are manipulated and changed by a program.
  - Properties such as precedence rules and parentheses determine how expressions become evaluated.

- **Statement:**
  - Statements such as conditional or iterative statements, determine how control flows from one segment of a program to another.

- **Declarative programming:**
  - It is an execution model that does not depend on statements, but nevertheless causes execution to proceed through a program.

- **Subprograms:**
  - Subprograms such as subprogram calls and coroutines, form a way to transfer control from one segment of a program to another.