Basic Programming Concepts

CS10001: Programming & Data Structures

Pallab Dasgupta
Professor, Dept. of Computer Sc. & Engg.,
Indian Institute of Technology Kharagpur
Some Terminologies

• **Algorithm / Flowchart**
  - Independent of the programming language.

• **Program**
  - A translation of the algorithm/flowchart into a form that can be processed by a computer.
  - Typically written in a high-level language like C, C++, Java, etc.
Variables and Constants

• Most important concept for problem solving using computers

• All temporary results are stored in terms of variables
  – The value of a variable can be changed.
  – The value of a constant do not change.

• Where are they stored?
  – In main memory.
• **How does memory look like (logically)?**
  
  – As a list of storage locations, each having a unique address.
  
  – Variables and constants are stored in these storage locations.
  
  – A variable is like a *bin*
    - The contents of the *bin* is the *value* of the variable
    - The variable name is used to refer to the value of the variable
    - A variable is mapped to a *location* of the memory, called its *address*
Memory map

Every variable is mapped to a particular memory address
Variables in Memory

<table>
<thead>
<tr>
<th>Instruction executed</th>
<th>Variable X</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X = 10$</td>
<td>10</td>
</tr>
<tr>
<td>$X = 20$</td>
<td>20</td>
</tr>
<tr>
<td>$X = X + 1$</td>
<td>21</td>
</tr>
<tr>
<td>$X = X \times 5$</td>
<td>105</td>
</tr>
</tbody>
</table>
Variables in Memory (contd.)

- **Instruction executed**
  - $X = 20$
  - $Y = 15$
  - $X = Y + 3$
  - $Y = X / 6$

<table>
<thead>
<tr>
<th>Variable</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3</td>
</tr>
</tbody>
</table>
Data Types

- **Three common data types used:**
  - **Integer**: can store only whole numbers
    - Examples: 25, -56, 1, 0
  - **Floating-point**: can store numbers with fractional values.
    - Examples: 3.14159, 5.0, -12345.345
  - **Character**: can store a character
Data Types (contd.)

- **How are they stored in memory?**
  - **Integer ::**
    - 16 bits
    - 32 bits
  - **Float ::**
    - 32 bits
    - 64 bits
  - **Char ::**
    - 8 bits (ASCII code)
    - 16 bits (UNICODE, used in Java)

*Actual number of bits vary from one computer to another*
Problem solving

- **Step 1:**
  - Clearly specify the problem to be solved.

- **Step 2:**
  - Draw flowchart or write algorithm.

- **Step 3:**
  - Convert flowchart (algorithm) into program code.

- **Step 4:**
  - Compile the program into object code.

- **Step 5:**
  - Execute the program.
Flowchart: basic symbols

- Computation
- Input / Output
- Decision Box
- Start / Stop
Contd.

Flow of control

Connector
Example 1: *Adding three numbers*

```
START

READ A, B, C

S = A + B + C

OUTPUT S

STOP
```
Example 2: *Larger of two numbers*

START

READ X, Y

IS X>Y?

YES

OUTPUT X

STOP

NO

OUTPUT Y

STOP
Example 3: \textit{Largest of three numbers}

```
START

READ X, Y, Z

IS X > Y?

\textbf{YES} \quad Max = X

\textbf{NO} \quad Max = Y

IS Max > Z?

\textbf{YES} \quad OUTPUT Max

STOP

\textbf{NO} \quad OUTPUT Z

STOP
```
Example 4: *Sum of first N natural numbers*

```
START
READ N
SUM = 0
COUNT = 1
SUM = SUM + COUNT
COUNT = COUNT + 1
IS COUNT > N?
NO SUM = SUM + COUNT
COUNT = COUNT + 1
YES OUTPUT SUM
STOP
```
Example 5: \( \text{SUM} = 1^2 + 2^2 + 3^2 + N^2 \)
Example 6: $SUM = 1.2 + 2.3 + 3.4 + \text{to N terms}$

START

READ N

SUM = 0
COUNT = 1

SUM = SUM + COUNT \*(COUNT + 1)

COUNT = COUNT + 1

IS COUNT > N?

NO

YES

OUTPUT SUM

STOP
Example 7: Computing Factorial

START

READ N

PROD = 1
COUNT = 1

PROD = PROD * COUNT

COUNT = COUNT + 1

IS COUNT > N?

NO

YES

OUTPUT PROD

STOP

Dept. of CSE, IIT KGP
Example 8: Computing $e^x$ series up to $N$ terms

1. START
2. READ $X$, $N$
3. TERM = 1
4. SUM = 0
5. COUNT = 1
6. SUM = SUM + TERM
7. TERM = TERM * X / COUNT
8. COUNT = COUNT + 1
9. IF COUNT > $N$ THEN YES ELSE NO
10. OUTPUT SUM
11. STOP
Example 8: Computing $e^x$ series up to 4 decimal places

```
START
READ X, N
TERM = 1
SUM = 0
COUNT = 1

SUM = SUM + TERM
TERM = TERM * X / COUNT
COUNT = COUNT + 1

IS TERM < 0.0001?

YES
OUTPUT SUM
STOP

NO

```
Example 10: *Roots of a quadratic equation*

\[ ax^2 + bx + c = 0 \]

*TRY YOURSELF*
Example 11: *Grade computation*

<table>
<thead>
<tr>
<th>Marks</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>≥ 90</td>
<td>Ex</td>
</tr>
<tr>
<td>89 ≥ Marks ≥ 80</td>
<td>A</td>
</tr>
<tr>
<td>79 ≥ Marks ≥ 70</td>
<td>B</td>
</tr>
<tr>
<td>69 ≥ Marks ≥ 60</td>
<td>C</td>
</tr>
<tr>
<td>59 ≥ Marks ≥ 50</td>
<td>D</td>
</tr>
<tr>
<td>49 ≥ Marks ≥ 35</td>
<td>P</td>
</tr>
<tr>
<td>34 ≥ Marks</td>
<td>F</td>
</tr>
</tbody>
</table>
Grade Computation (contd.)

START

READ MARKS

MARKS ≥ 90?

YES

OUTPUT “Ex”

STOP

NO

MARKS ≥ 80?

YES

OUTPUT “A”

STOP

NO

MARKS ≥ 70?

YES

OUTPUT “B”

STOP

NO

A

Dept. of CSE, IIT KGP