CS1111: Problem Solving using Computers

Rupesh Nasre.

Mentor TAs: Ahmed, Vivek, Vimala, Akash,
Kankan, Rahul, Swati, Akshay, Ashok, Keshav

IIT Madras
May 2022

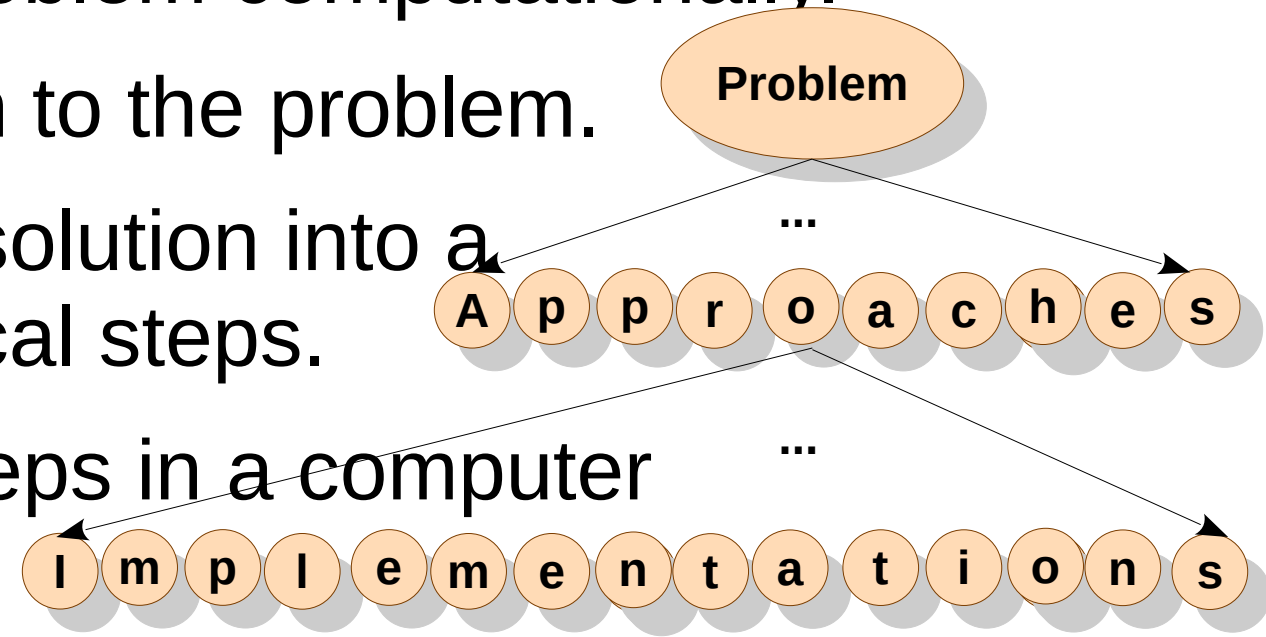
Placement in Computer Science

- **CS1111: Problem Solving and Coding**
- CS1200: Proofs, Counting
- CS2200: Computation Theory
- CS2300: Overview of Digital World
- CS2600: Hardware
- CS2700: Efficient Implementation
- CS2800: Algorithms
- CS3100: Ways of Programming
- CS3300: Translation (Programmer and Machine)
- CS3500: Resource Management (User and Machine)

CS1111 is a foundational subject feeding into all the other CS courses.

Learning Outcomes

- Model a given problem computationally.
- Identify a solution to the problem.
- Decompose the solution into a sequence of logical steps.
- Implement the steps in a computer program.
- Solve the problem with the program.
  - Iterate through the solution as required.



Our first problem: Make tea.

1. Take tea-powder. // how much?
2. Take sugar. // where is it?
3. Take milk. // what if I don't have milk?
4. Boil together. // for how long?
5. Tea is ready!

Even for intellectuals such as humans, we need more information.
For dumb machines such as computers, we need to be **precise**.

Programming is about precise understanding;
so precise that even a machine should be able to follow.

Preparation Time: 2 minutes

Cooking Time: 10 minutes

Serves: 2 servings (2 cups)

Print Recipe

Cooking Measurements

Data

Ingredients:
1 cup (250 ml) Milk
2 teaspoons Tea Powder
1/4 cup (approx. 60 ml) Water
3 teaspoons Sugar

Your laptop is unlikely to be able to make tea. But then ...

Algorithm

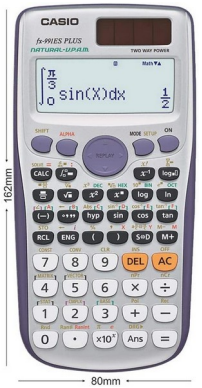
Directions:

1. Boil water in a saucepan.
2. Add sugar and tea powder in it and boil it for 3-4 minutes on medium flame.
3. Add milk and boil it over medium flame for 6-7 minutes or until bubble starts to rise. You will see the change in color of the tea from milky shade to brown shade when it is ready.
4. Turn off the gas and strain tea in cups.

Is your tea vending machine a computer?

What if it can give you tea, coffee, milk, hot water, ...?

How about a calculator?



A computer is **programmable**.

A vending machine or a calculator are not. They can perform only *pre-programmed* computation.

Then how about your iPads or smart phones?

amazon.com
computerhope.com

Week	Problems	Tools
0	Solve equations, find weighted sum.	Data types, expressions, assignments
1	Find max, convert marks to grade.	Conditionals, logical expressions
2	Find weighted sum for all students.	Loops
3	Encrypt and decrypt a secret message.	Character arrays
4	Our first game: Tic-tac-toe	2D arrays
5	Making game modular, reuse.	Functions
6	Find Hemachandra/Fibonacci numbers.	Recursion
7	Encrypt and decrypt many messages.	Dynamic memory, pointers
8	Maintain student records.	Aggregate data types
9	Search and sort student records.	Searching and sorting algorithms
A	Reduce memory wastage.	Linked lists
B	Implement token system in banks.	Queues
C	IRCTC-like ticket booking system	File handling
D	Putting it all together	All the above

Logistics

- Course credits: 3-0-0-3-6-12
- if (date <= May 12) {

  theoryAt(Mon 13 + 17, Wed 17, Thu 15 + 17, Fri 14);

  labAt(Thu 9, Fri 9);

  } else {

  theoryAt(Mon 13 + 14, Wed 14, Thu 10 + 14, Fri 9);

  labAt(Thu 15, Fri 15);

  }
- We will use replit platform for the labs.

Logistics

- **Evaluation**
  - 56% labs + 15% midsem + remaining% endsem
  - Every lab is evaluated.
  - Attendance: Standard institute rules apply.
  - Midsem and endsem dates will be populated on the course webpage (along with slides and codes).
- **Moodle**
  - Will be used as a communication mechanism.
  - Your responsibility to subscribe to it.
  - Join here.

To get the MOST out of this course

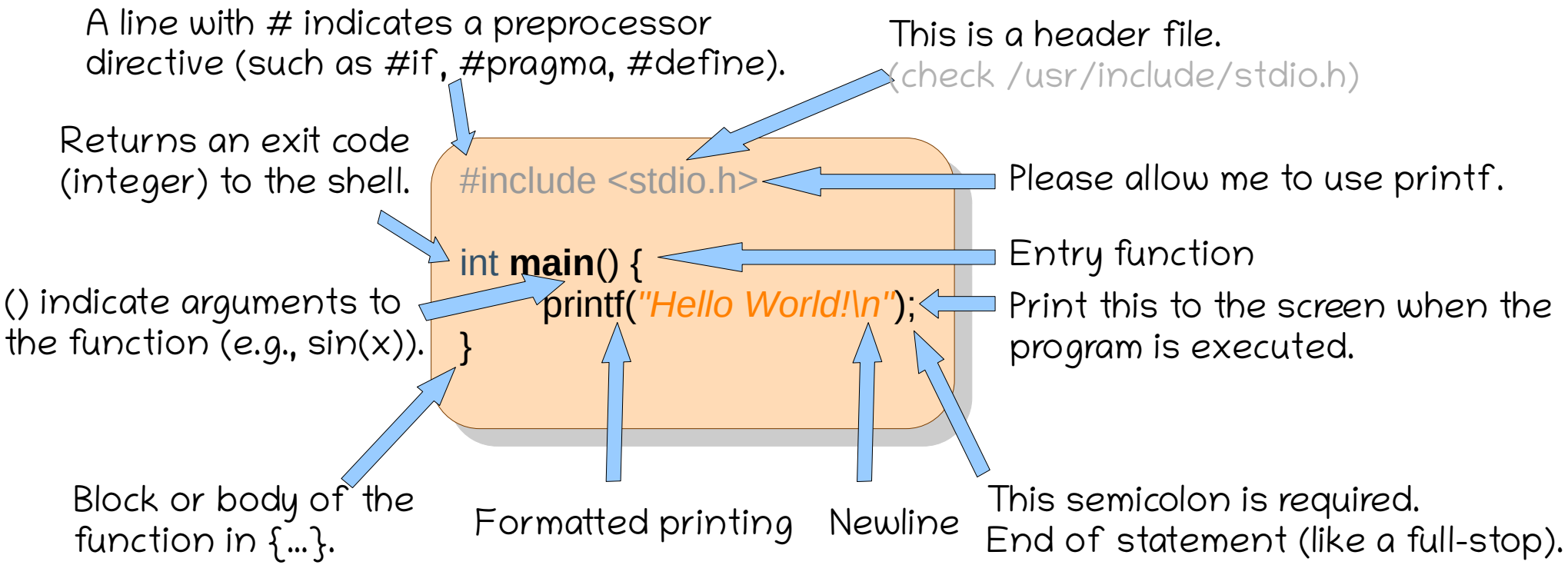
- Keep hands away from WhatsApp.
- Solve questions during classwork.
  - Keep a copy with you. Take notes.
- Ask questions (others also haven't understood).
  - Do not let a few dominate the discussion.

First Program

print “Hello World!“

- Unfortunately, this is Tamil for a Bengali person.
- And our mother-tongue is C. So we will have to follow the C syntax.
- Where do you write this program?
  - On Linux: text editor, vi, VS code, nano, sublime, ...
  - On replit

Hello World!




```
#include        <stdio.h>
int main(    )
{printf   (   "Hello World!\n")
  ;

}
```

```
main(){}
```

Whitespace can be added freely (almost).
Whitespace means space, tab, newline.

Smallest C program

A Small Problem

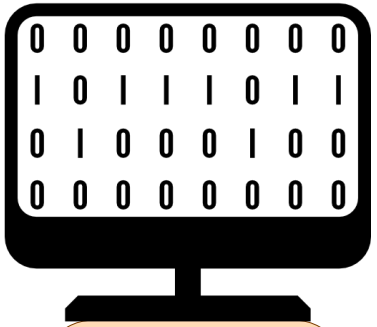
```
#include <stdio.h>

int main() {
        printf("Hello World!\n");
}
```

- This English-like program can be understood by humans, but not by machines – such as your laptop.
- Computers understand only 0 and 1.
- How about writing our code in binary?
  - Possible, but not very motivating.
- Is it possible to write in C and the machine reads binary?

C → Compiler → Machine code

```
#include <stdio.h>

int main() {
        printf("Hello World!\n");
}
```

Tamil Bengali



```
11000110
01010011
01010100
11101000
11111110
```

- We will use a translator!
- On replit and your Linux laptops, the compiler is gcc.

```
$ gcc hello.c
$ a.out
Hello World!
$
```

Command to compile. Translates .c file to a.out.
Run your program (or execute it).
Output of your program.
Command prompt to type the next command.

gcc is also a big program written by many people, **such as you**.
So are firefox, chrome, minesweeper, powerpoint, Windows OS, Android OS, ...

```c
#include <stdio.h>

int main() {
        printf("Hello World!\n");
}
```

```c
#include <stdio.h>

int main() {
        printf("Bye World!\n");
}
```

```c
#include <stdio.h>

int main() {
        printf("printf(printf)\n");
}
```

```c
#include <stdio.h>

int main() {
        printf("Hello ");
        printf("World!\n");
}
```

```c
#include <stdio.h>

int main() {
        printf("Hello "
                "World!\n");
}
```

```c
#include <stdio.h>

int main() {
        printf("Hello \
World\n");
}
```

```c
#include <stdio.h>

int main() {
        printf("3*5 = 15\n");
}
```

```c
#include <stdio.h>

int main() {
        printf(3*5);
}
```

```c
#include <stdio.h>

int main() {
        printf(TN's CM);
}
```

3*5 = 15

Compiler issues a warning, but compiles to a.out.
a.out prints
Segmentation fault (core dumped)

Compiler issues a warning and an error.
What does a.out print?

We need to understand printf a little better.

printf

printf here took two arguments
"3*5 = %d\n" and 3*5

The first argument in double-quotes is called a **format-string**.

This format-string is our bus without people, but with their placeholders.

Placeholder for decimal integer.

```
#include <stdio.h>

int main() {
        printf("3*5 = %d\n", 3*5);
}
```

```
3*5 = 15
```

Placeholders, to be replaced by people.

printf

```
#include <stdio.h>

int main() {
        printf("3*5 = %d, a-z are %d letters, 5 is 50%% of %d\n", 3*5, 26, 10);
}
```

```
3*5 = 15, a-z are 26 letters, 5 is 50% of 10
```

Placeholders, to be replaced by people.

Future Connect:
printf permits variable number of arguments.

printf

There are more **format specifiers**, which we will study soon.

```
#include <stdio.h>

int main() {
        printf("3*5 = %d, a-z are %d letters, 5 is 50%% of %d\n", 3*5, 26, 10);
}
```

```
3*5 = 15, a-z are 26 letters, 5 is 50% of 10
```

- What if there is a mismatch in the number of placeholders and the number of arguments?
  - For a **correct program**, the two should match.
  - You can play around with these numbers to know the behavior of the compiler / runtime, but it would not fetch you much w.r.t. the application semantics.
- Why does C have such a cryptic way for simple printing?

Classwork: Find outputs.

```
#include <stdio.h>

int main() {
        printf("3*5\n = 15\n");
}
```

```
3*5
 = 15
```

```
#include <stdio.h>

int main() {
        printf("3*5 = %d\n", 16);
}
```

```
3*5 = 16
```

```
#include <stdio.h>

int main() {
        printf("15 is 15% of 100\n");
}
```

```
15 is 1531263471240f 100
```

printf

There are more **format specifiers**, which we will study soon.

```
#include <stdio.h>

int main() {
        printf("3*5 = %d, a-z are %d letters, 5 is 50%% of %d\n", 3*5, 26, 10);
}
```

Format specifier	Meaning
%d	Decimal integer
%o	Octal integer
%c	Character
%f	Real number
...	...

Data types

Some of these are provided by C. Others can be created by us.

Why do we need data types?

- Numbers are of different types (number of students vs. height).
- Text vs. numbers vs. roll number
- Academic record vs. bank account transactions
- ...

printf Format Specifiers

Format specifier	Meaning	C type	Constant	Examples
%d	Decimal integer	int	99, 0, -1, 600036	Number of books, pincode, number of classes attended
%o	Octal integer	int	010, 071	Same in octal
%c	Character	char	'a', 'C'	First letter of name, grade in CS1111, blood group
%f	Real number	float	-2.345, 1.0e10	Height, PI, percentile
%s	String	char [ ]	"Hello World!", "CS21B018"	Name, commands, arbitrary text
%p	Pointer	Type *	0xFF112233	Address of a variable
%i, %u	Similar to %d (works for positive values)			
%x, %X	Hexadecimal (0-9, a-f or A-F)			
%e,%E,%g	Similar to %f, but for scientific notation or fixed-precision			
%ld, %li	Long integer (larger values)			
%lf, %Lf	Double (long float), Long double			
%hi, %hu	Short integer (smaller values)			
%n	Number of characters printed by this printf so far.			
%%	Character %			

```
printf("%s of %s\n",
        "Summer", "69");
```

```
printf("%s has %d students\n",
        "CS1111", 87);
```

```
printf("The value of PI");
printf(" is %f", 3.1428);
```

```
printf("%d", 3);
printf(" idiots");
```

```
printf("name = %s\n", "Khan");
printf("age = %d\n", 20);
printf("height = %lf ft\n", 5.8);
printf("weight is 50kg\n");
```

```
printf("CS%d is %s course\n",
        1111, "foundational");
```

```
printf("Hexadecimal has \
%X symbols.\
\n", 16);
```

```
printf("32 1s is %x in hex \
and %o in octal\n",
0xFFFFFFFF, 0xFFFFFFFF);
```

```
printf("%cS%x\n", 'C', 4369);
```

This was output. Does C have a capability to take input?

```
Hello, which course is this?
1111
Hi! Welcome to CS1111.
```

We want this to be typed by the user.

We ~~need to~~ understand printf a little better.

How does our brain remember?

It stores the information in memory cells.

Can we also do the same?

But which cell to access?

Hmm... Let's name the cells.

printf("Hello, which course is this?\n");

What user enters here should appear below.

printf("Hi! Welcome to CS%d\n", **???**);

How about this?

We know how to print this.

Hello, which course is this?
1111
Hi! Welcome to CS1111.

We want this to be typed by the user.

How does our brain remember?

It stores the information in memory cells.

Can we also do the same?

But which cell to access?

Hmm... Let's name the cells.

```
printf("Hello, which course is this?\n");

What user enters is stored in cell cell1.

printf("Hi! Welcome to CS%d\n", cell1);
```

Where is this cell1 stored?

Inside your computer.

But where?

Well, in memory.

We call it random access memory.

How about this?

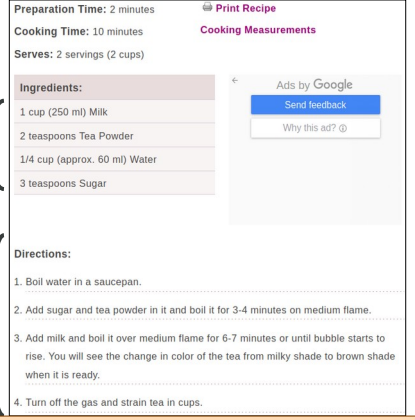
We know how to print this.

```
Hello, which course is this?
1111
Hi! Welcome to CS1111.
```

We want this to be typed by the user.

Data

Algorithm




```
int cell1;
printf("Hello, which course is this?\n");
scanf("%d", cell1);
printf("Hi! Welcome to CS%d\n", cell1);
```

This code doesn't compile.

gcc says it doesn't know **cell1**.

Let's compile and run it.

```
printf("Hello, which course is this?\n");

What user enters is stored in cell cell1.

printf("Hi! Welcome to CS%d\n", cell1);
```

Where is this cell1 stored?

Inside your computer.

But where?

Well, in memory.

We call it random access memory.

How about this?

We know how to print this.

```
Hello, which course is this?
1111
Hi! Welcome to CS1111.
```

We want this to be typed by the user.

Hello, which course is this?
1111
Segmentation fault (core dumped)

```
int cell1;
printf("Hello, which course is this?\n");
scanf("%d", cell1);
printf("Hi! Welcome to CS%d\n", cell1);
```

Let's compile and run it.

```
printf("Hello, which course is this?\n");

What user enters is stored in cell cell1.

printf("Hi! Welcome to CS%d\n", cell1);
```

Where is this cell1 stored?

Inside your computer.

But where?

Well, in memory.

We call it random access memory.

How about this?

We know how to print this.

Hello, which course is this?
1111
Hi! Welcome to CS1111.

We want this to be typed by the user.

A New Problem

- Say you want to get your room/house painted.

- Which of the following whatsapp messages would help a painter reach and paint your house?
  - Hi, my house color is white.
  - Hi, my house is around 1500 sq ft.
  - "Hi, I have %d members in my house.", 4

  Give some information but the painter can't change the color.

  - Hi, my house address is 670, New Nandanvan.

    Allows the painter to go and change the color.
    In addition, the painter can get the other information also!

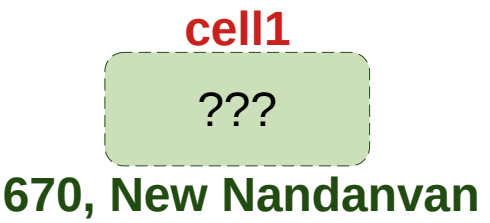
Coming back to the old problem

```
int cell1;
printf("Hello, which course is this?\n");
scanf("%d", cell1);
printf("Hi! Welcome to CS%d\n", cell1);
```

Informs scanf of the value in **cell1**, but does not allow scanf to change it.

We need to send address of **cell1**.

```
int cell1;
printf("Hello, which course is this?\n");
scanf("%d", 670, New Nandanvan);
printf("Hi! Welcome to CS%d\n", cell1);
```



- Hi, my house address is 670, New Nandanvan.

  Allows the painter to go and change the color.
  In addition, the painter can get the other information also!

Coming back to the old problem

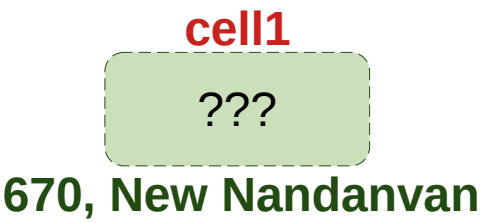
What if I I pass cell1 to scanf?

```
int cell1;
printf("Hello, which course is this?\n");
scanf("%d", cell1);
printf("Hi! Welcome to CS%d\n", cell1);
```

Informs scanf of the value in **cell1**, but does not allow scanf to change it.

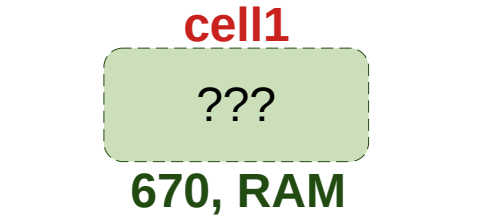
We need to send address of **cell1**.

```
int cell1;
printf("Hello, which course is this?\n");
scanf("%d", 670, New Nandanvan);
printf("Hi! Welcome to CS%d\n", cell1);
```



What if I pass &cell1 to printf?

```
int cell1;
printf("Hello, which course is this?\n");
scanf("%d", &cell1);
printf("Hi! Welcome to CS%d\n", cell1);
```



Note that printf is interested in the value, and not in changing it.

Coming back to the old problem

Hello, which course is this?
1111
Hi! Welcome to CS1111

address

value

cell1 syntactic sugar

1111 passed to printf

670 passed to scanf

Hello, which course is this?
6023
Hi! Welcome to CS6023

cell1

6023

424

Hello, which course is this?
CS1111
Hi! Welcome to CS0

cell1

0

35213944

```
int cell1;
printf("Hello, which course is this?\n");
scanf("%d", &cell1);
printf("Hi! Welcome to CS%d\n", cell1);
```

cell1

???

670, RAM

```
int year;
scanf("%d", &year);
printf("%s of %d\n",
        "Summer", year);
```

```
int nstuds;
scanf("%d", &nstuds);
printf("%s has %d students\n",
        "CS1111", nstuds);
```

```
float pi;
printf("Value of pi?");
scanf("%f", &pi);
printf("The value of PI");
printf(" is %f", pi);
```

```
short nidiots;
scanf("%d", &nidiots);
printf("%d", nidiots);
printf(" idiots");
```

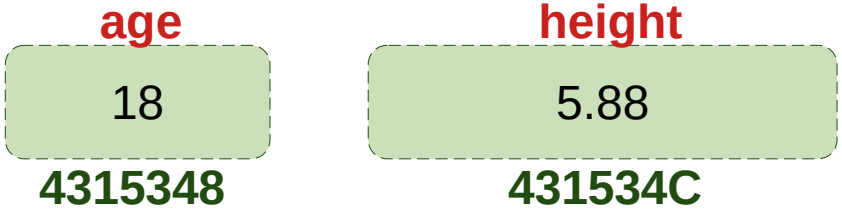
```
int age; double height;
scanf("%d", &age);
printf("age = %d\n", age);
scanf("%lf", &height);
printf("height = %lf ft\n", height);
printf("weight is 50kg\n");
```

```
long int course;
scanf("%ld", &course);
printf("CS%d is %s course\n",
        course, "foundational");
```

```
int base16;
scanf("%x", &base16);
printf("Hexadecimal has \
%X symbols.\
\n", base16);
```

```
int max32;
scanf("%x", &max32);
printf("32 1s is %x in hex \
and %o in octal\n",
max32, max32);
```

```
char c;
int num;
scanf("%c", &c);
scanf("%d", &num);
printf("%cS%x\n", c, num);
```



Problem: Find age from birth year.

Algorithm and Implementation

```
// create cell birthyear
int birthyear;
// take input from user in cell birthyear
scanf("%d", &birthyear);
// create cell age
int age;
// calculate (2022 – birthyear) and store in age.
age = (2022 – birthyear);
// output age
printf("Your age is %d\n", age);
```

Comment

Assignment

Modulus or remainder

```
int A, b, C;
A = 0;
b = 1 - A;
C = A - b;
```

A	b	C
0	1	-1

```
float celcius, farenheit;
scanf("%f", &celcius);
farenheit = 9*celcius/5 + 32;
```

```
scanf("%d", &num);
mod5 = num % 5;
printf("%d mod 5 is %d\n",
        num, mod5);
```

Problem: Find your team number.

```
// create cell for last two digits of roll number
int rollnumber;
// take input from user in cell rollnumber
scanf("%d", &rollnumber);
// create cell for team number
int teamnumber;
// find rollnumber / 10 + 1 and store in teamnumber
teamnumber = ((rollnumber - 1) / 10 + 1);
// output team number
printf("Your team is Team %d\n", teamnumber);
```

Extent the program for full roll number.

Expect last two digits alone.

```
15
Your team is Team 2
78
Your team is Team 8
20
Your team is Team 2
```

Testing helps find bugs.

- Here is your replit team mapping.
  - Team 1: Roll numbers CS21B001 – 10
  - Team 2: Roll numbers CS21B011 – 20
  - Team 9: Roll numbers CS21B081 -- 86

Given a team number, find the first and the last roll numbers in that input team. (assume 90 students)

Problem: Find team members.

As a good programming practice
- Avoid implicit type conversion.
- Avoid constants such as 65.

```
// create data
char teamid;
int startroll, endroll;
// take input
scanf("%c", &teamid);
// find the roll numbers
startroll = ((int)teamid – 'A') * 10 + 1;
endroll = startroll + 9;
// output
printf("%d to %d\n", startroll, endroll);
```

Characters are stored as integers.
'A' is 65, 'B' is 66, ...
(teamid + 5) is allowed in C.

A
1 to 10
I
81 to 90
C
21 to 30

- Here is your replit team mapping.
  - Team **A**: Roll numbers CS21B001 – 10
  - Team **B**: Roll numbers CS21B011 – 20
  - Team **I** : Roll numbers CS21B081 -- 90

Given a team id, find the first and the last roll numbers in that team. (assume 90 students)

Problem: Find endsem percentage.

```
// create data
int labs, midsem, endsem;
// take input
scanf("%d%d", &labs, &midsem);
// do computation
endsem = (100 – (labs + midsem));
// output
printf("Endsem %% is %d\n", endsem);
```

labs	midsem	remaining
56	15	29

```
endsem = (100 – labs – midsem);
```

```
int nonendsem;
nonendsem = labs + midsem;
endsem = 100 – nonendsem;
```

```
int remaining;
remaining = 100;
scanf("%d", &labs);
remaining = remaining – labs;
scanf("%d", &midsem);
remaining = remaining – midsem;
printf("Endsem %% is %d\n", remaining);
```

- 56% labs + 15% midsem + remaining% endsem

```
56
15
29
```

Problem: Find sum.

```
int n;
// read n
scanf("%d", &n);
// compute sum
int sum = n * (n + 1) / 2;
// print sum
printf("Sum of first %d numbers is %d\n", n, sum);
```

$$\Sigma n = 1 + 2 + 3 + \ldots + n$$

$$= n * (n + 1) / 2$$

$$\Sigma 2^i = 1 + 2 + 4 + 8 + \ldots \text{ n terms}$$

$$= 2^n - 1$$

Needs pow() function or a loop.
pow(x, y) returns x^y.
Does your code compile?

Problem: Find probability.

```
int nCards = 52;
int nKings = 4;
int nSpades = 13;
int nKingAndSpade = 1;
int nNonkingNonspade = nCards -
      (nKings + nSpades - nKingAndSpade);
float prob = nNonkingNonspade / nCards;
printf("Probability of nonking, nonspade is %f\n",
      prob);
```

Find the issue with this code.

A card is drawn at random from a deck of well-shuffled cards. Find the probability of it being neither a king nor a spade.

Problem: Find the line.

```
float x1, y1, x2, y2;
scanf("%f%f%f%f", &x1, &y1, &x2, &y2);
float m = (y2 – y1) / (x2 – x1);
float c = (y1 – m*x1);
printf("Equation of the line is y = %.2fx + %.2f\n", m, c);
printf("Equation of the line is y = %gx + %g\n", m, c);
printf("Equation of the line is y = %.2ex + %.2e\n", m, c);
```

```
3.2   3   9.6   5
Equation of the line is y = 0.31x + 2.00
Equation of the line is y = 0.3125x + 2
Equation of the line is y = 3.12e-01x + 2.00e+00
```

Given two points on a line, find its equation in $y = mx + c$ format.

Future Connect:
Replacing &name1 with name1 also works.

Problem: Print tabular.

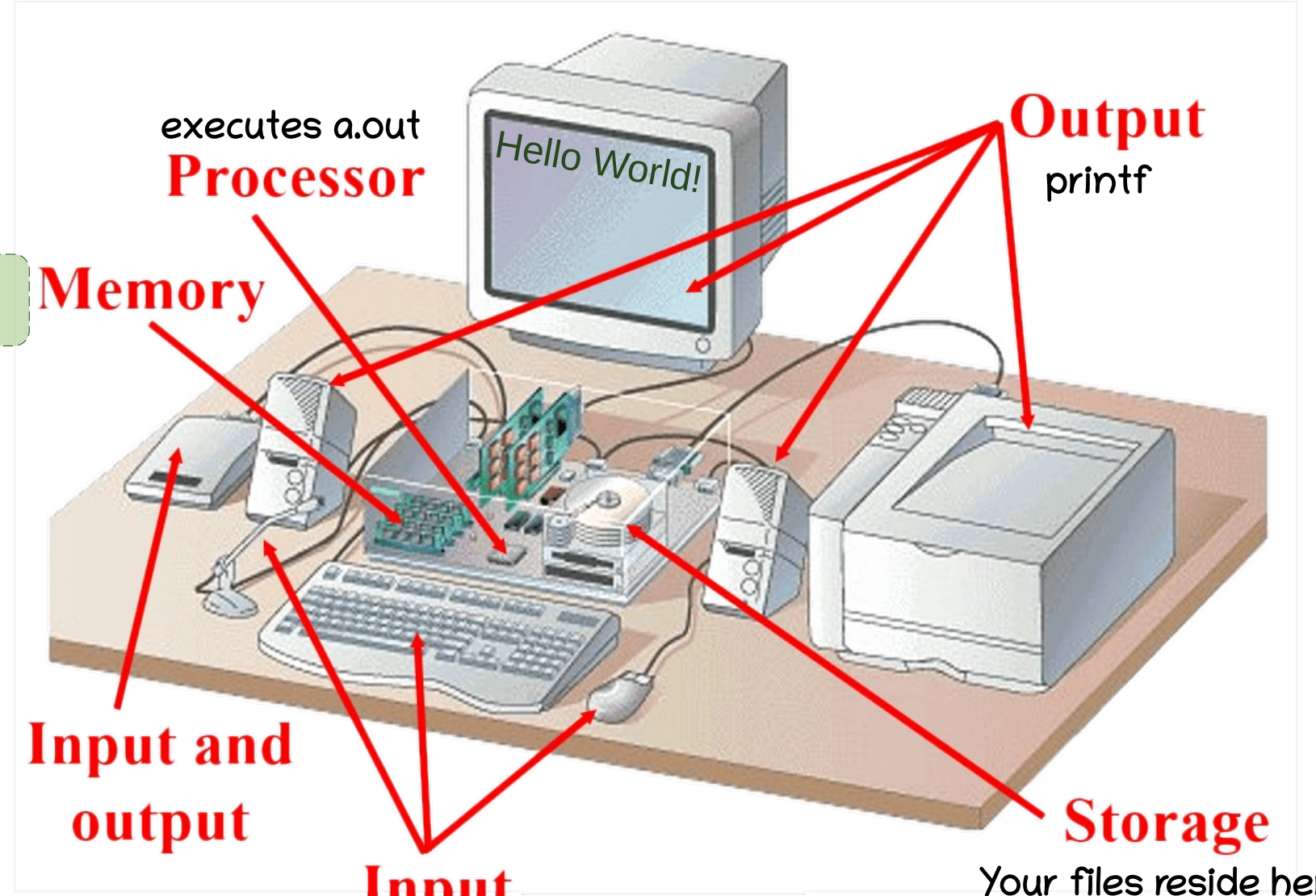
```
char name1[20], name2[20], name3[20];
int m11, m12, m13, m21, m22, m23, m31, m32, m33;

scanf("%s%d%d%d", &name1, &m11, &m12, &m13);
scanf("%s%d%d%d", &name2, &m21, &m22, &m23);
scanf("%s%d%d%d", &name3, &m31, &m32, &m33);

int t1, t2, t3;
t1 = m11 + m12 + m13;
t2 = m21 + m22 + m23;
t3 = m31 + m32 + m33;

printf("%-12s%5d%5d%5d = %5d\n", name1, m11, m12, m13, t1);
printf("%-12s%5d%5d%5d = %5d\n", name2, m21, m22, m23, t2);
printf("%-12s%5d%5d%5d = %5d\n", name3, m31, m32, m33, t3);
```

char array or string

```
Rajesh         1   43   43 =    87
SomeshSingh   23   55    6 =   144
JK            21   21   21 =    63
```

Read names and marks of three students and print the names and total in a table.

Computer System



age

18

4315348

All C Keywords

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

Summary

- Hello World!
- Formatted input, output
- Problem Solving with assignments