1.1 & 1.2
Introduction

Human-based computing

- In the Second World War, one of the challenges for each side was to try to crack the codes used by the opposition.
- This became urgent for the British as the German Uboat fleet was decimating British convoys in the Atlantic.
- The Admiralty desperately needed to know where the U-boats were, and this could only be achieved by intercepting their communications.
- In Britain the code-breaking work was done at Bletchley Park, where originally teams of people worked to decipher messages.
- Many did not understand the purpose of their work, as they only worked on small parts and did not see the bigger picture.
- They just did the calculations needed by blindly following the instructions they were given.

Machine based computing

- The instructions being followed were devised by a team of Mathematicians led by Alan Turing: the original human computer programmers.
- They of course had to understand what was going on to devise the instructions.
- However, the mass of communications that were being intercepted meant that the teams of people could not work fast enough.
- Gradually machines were designed that could do some of the routine parts.
- These machines were the precursors to the modern computer (Hodges, 1985).
- Machine-based computers thus just took over the things that human-based computers had previously done.
- Their speed and accuracy mean they can tackle bigger problems and do them faster, but the essence of what they do remains the same.

Data around us

- Our world is made of all kinds of objects or things that we like to be informed of.
- Each object is described by a number of attributes (*characteristics*).
- Each attribute can have a value and that value is data.

Figure: An object's characteristics become data

✓ *In the given example, how many types of data do you see?*

Data and Information

- Data- Raw unorganized facts that need to be processes.
- It can be something simple, seemingly random and useless until it is organized.

  Each student's text score is one piece of data.

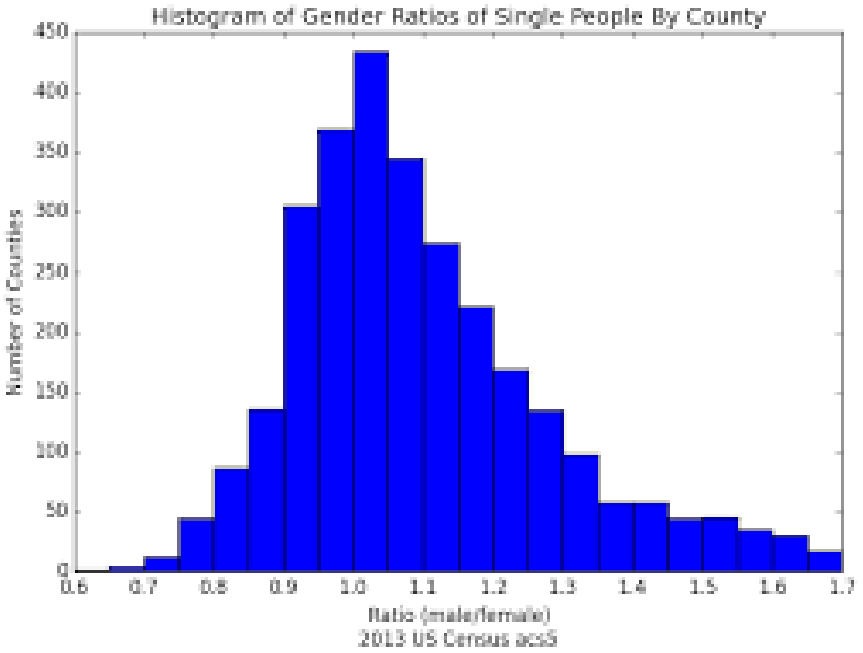
- Information-Data when processed, organized, structured or presented in a given context so as to make it useful.

  Average score of a class department derived from the data.

✓ *Can you give some examples of data and associated information?*

Census Report (Total population, Literacy Rate, Male / Female ratio etc.)

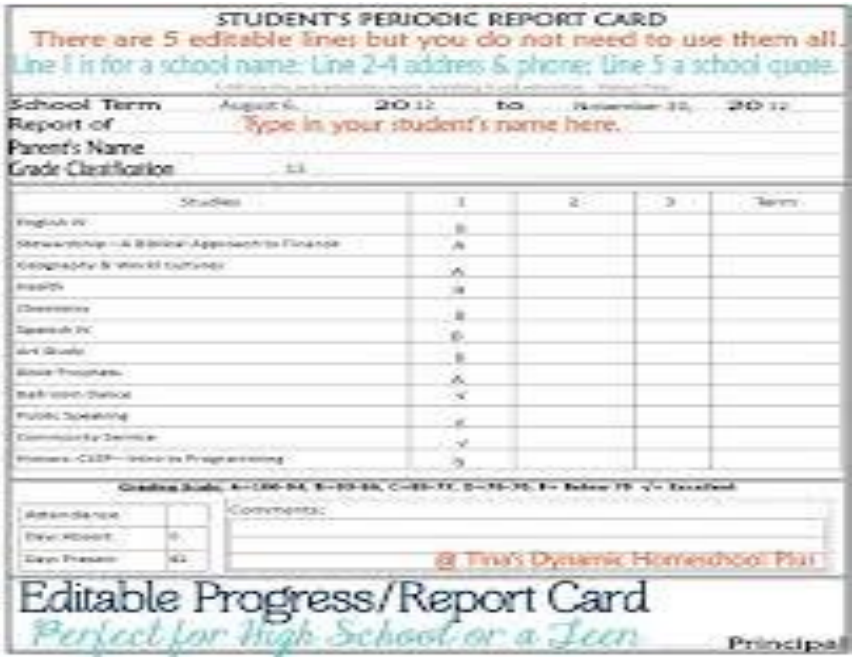
- ✓ Census data is used to get report/information about total population of a country and literacy rate etc.



Result Cards of Individual Students, Result sheets of a particular Class

- ✓ In examination system collected data (obtained marks in each subject) is processed to get total obtained mark of a student.

- ✓ Total obtained mark is Information. It is also used to prepare result card of a student.



Merit List

- ✓ After collecting admission forms from candidates, merit is calculated on the basis of obtained marks of each candidate.
- ✓ Normally, percentage of marks obtained is calculated for each candidate.
- ✓ Now all the candidates names are arranged in descending order by percentage.
- ✓ This makes a merit list.
- ✓ Merit list is used to decide whether a candidate will get admission in the college or not.

UHS 1st MBBS Merit List Highest and Lowest Merit of all Medical Colleges

Compiled by : Team Educational Blog

www.Edu.Apnafort.com

College Name	Highest Merit	Lowest Merit	Total Open Merit Seats
KEMU	95.7143	90.3529	302
NMC	93.6111	88.9172	80
SIMS	89.8723	88.1835	191
FJMC	92.3305	87.3657	206
SKZMDC	85.2037	83.4277	90
MC	89.4442	87.0848	100
MC	92.5996	87.0970	280
MC	94.0961	86.5835	297
MC	95.3965	86.5048	287
JMC	94.8026	86.3585	273
MC	87.4095	86.2881	100
MC	91.5165	86.0262	79
MC	91.5827	85.8559	90
MC	86.9015	85.7874	100
MC	88.6970	85.2874	126
MSMC	88.3506	85.2589	100
MC	87.8390	85.6451	100
		85.6411	Total: 3022

For Latest UHS MBBS & BDS Merit Lists Visit www.Edu.ApnaFort.com

There is more to technology than mere social media & networking

Figure: Technology enabled connectedness through social media

✓ *Can you name some (software) technologies that have impacted you?*

Software - product of critical thinking skills

Figure: The world of Software

- Along with the power of computing, software too became versatile and powerful.
- There are plenty of software out there; Today software is synonymous with technology.
- Software technologies were possible due to both power of computing and critical thinking skills.
- This course is about computational thinking skills.

- Computational Thinking is a term coined by Jeannette M. Wing of CMU.

- Communications of ACM article authored by her introduced the tenets of Computational Thinking.

- Computational Thinking is all about the ways to think like a computer scientist.

What is computer science all about?

Computer Science is the study of how to represent and manipulate the information inherent to natural, engineered, social and artistic systems.

Viewpoint | Jeannette M. Wing

Figure: Jeannette's Article

Computational Thinking

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.

Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of tackling alone. Computational thinking confronts the riddle of machine intelligence: What can humans do better than computers? and What can computers do better than humans? Most fundamentally it addresses the question: What is computable? Today, we know only parts of the answers to such questions.

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability. Just as the printing press facilitated the spread of the three Rs, what is appropriately incestuous about this vision is that computing and computers facilitate the spread of computational thinking.

Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science.

Having to solve a particular problem, we might ask: How difficult is it to solve? and What's the best way to solve it? Computer science rests on solid theoretical underpinnings to answer such questions precisely. Stating the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We must consider the machine's instruction set, its resource constraints, and its operating environment.

In solving a problem efficiently, we might further ask whether an approximate solution is good enough, whether we can use randomization to our advantage, and whether false positives or false negatives are allowed. Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation.

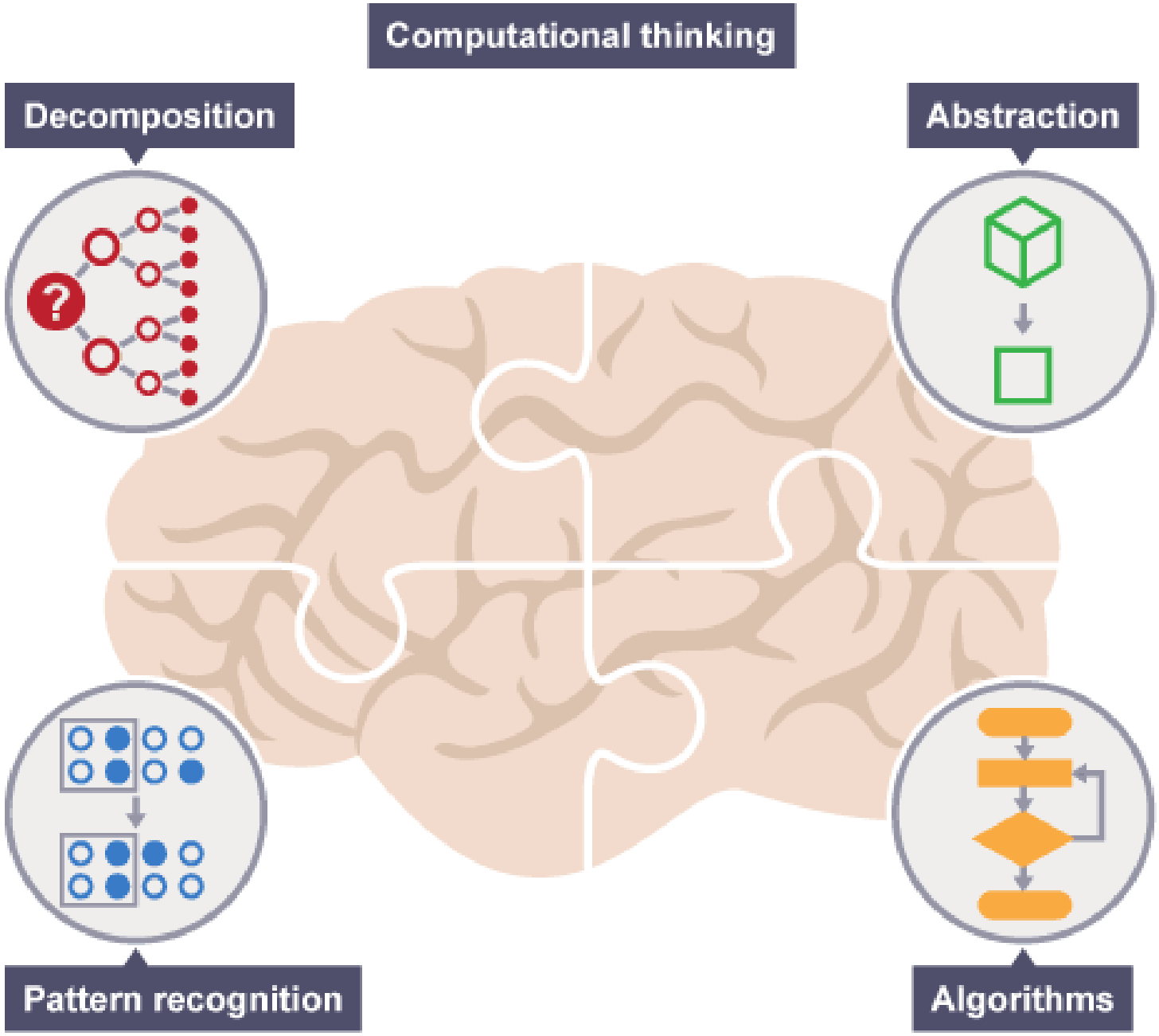
Computational thinking is thinking recursively. It is parallel processing. It is interpreting code as data and data as code. It is type checking as the generalization of dimensional analysis. It is recognizing both the virtues and the dangers of aliasing, or giving someone or something more than one name. It is recognizing both the cost and power of indirect addressing and procedure call. It is judging a program not just for correctness and efficiency but for aesthetics, and a system's design for simplicity and elegance.

Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system. It is separation of concerns. It is choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable. It is using invariants to describe a system's behavior succinctly and declaratively. It is having the confidence we can safely use, modify, and influence a large complex system without understanding its every detail. It is

LISA HANEY

Computational Thinking

- Computers can be used to help us solve problems.
- However, before a problem can be tackled, the problem itself and the ways in which it could be solved need to be understood.
- Computational thinking allows us to do this.
- Computational thinking allows us to take a complex problem, understand what the problem is and develop possible solutions.
- We can then present these solutions in a way that a computer, a human, or both, can understand.


Computational thinking
Decomposition
Abstraction
Pattern recognition
Algorithms

The four cornerstones

- **Decomposition** - breaking down a complex problem or system into smaller, more manageable parts
- **Pattern recognition** – looking for similarities among and within problems
- **Abstraction** – focusing on the important information only, ignoring irrelevant detail
- **Algorithms** - developing a step-by-step solution to the problem, or the rules to follow to solve the problem
- Each cornerstone is as important as the others.
- They are like legs of a table - if one leg is missing, the table will probably collapse.
- Correctly applying all four techniques will help when programming a computer.

Programs

- Human based computation
  - Human beings were also given programs to follow:
    - The instructions were just given in a language the humans could follow.
    - When you do a long multiplication you are doing exactly this, following step-by-step rules you wrote as a child.
- Computer programs are just step-by-step instructions for doing a given task, written in computer programming languages.

Algorithms

- These step-by-step instructions are algorithms.
  - An algorithm consists of the actions to do and the order to do them in.
- Algorithms occur all over the place
  - the instructions you learnt about how to do long division, find prime numbers, bake, brush your teeth etc…..
  - the instructions of how to put together a DIY bookcase
  - A set of instructions of how to solve a Rubik's Cube
    - *In fact there are a whole series of puzzles that involve devising algorithms to solve them of which a Rubik's cube is one of the most complex.*

Computational Thinking in Practice

- Before computers can be used to solve a problem, the problem itself and the ways in which it could be resolved must be understood.
- Computational thinking techniques help with this.
- Thinking computationally is not **programming**.
- It is not even thinking like a computer, as computers do not, and cannot, think.
- Simply put, programming tells a computer what to do and how to do it.
- **Computational thinking enables you to work out exactly what to tell the computer to do.**

What CT is and is'nt?

- ✓ Conceptualizing, not programming.
- ✓ Fundamental, not rote skill.
- ✓ A way that humans, not computers, think.
- ✓ Complements and combines mathematical and engineering skills.
- ✓ Ideas not artifact.
- ✓ For everyone, everywhere.

Planning to meet friends

- If you agree to meet your friends somewhere you have never been before, you would probably plan your route before you step out of your house.

- You might consider the routes available and which route is 'best' - this might be the route that is the shortest, the quickest, or the one which goes past your favourite shop on the way.

- You'd then follow the step-by-step directions to get there.

In this case, **the planning part is like computational thinking,** and **following the directions is like programming.**

- Being able to turn a complex problem into one we can easily understand is a skill that is extremely useful.

- In fact, it's a skill you already have and probably use every day.

Go out with a Group of friends

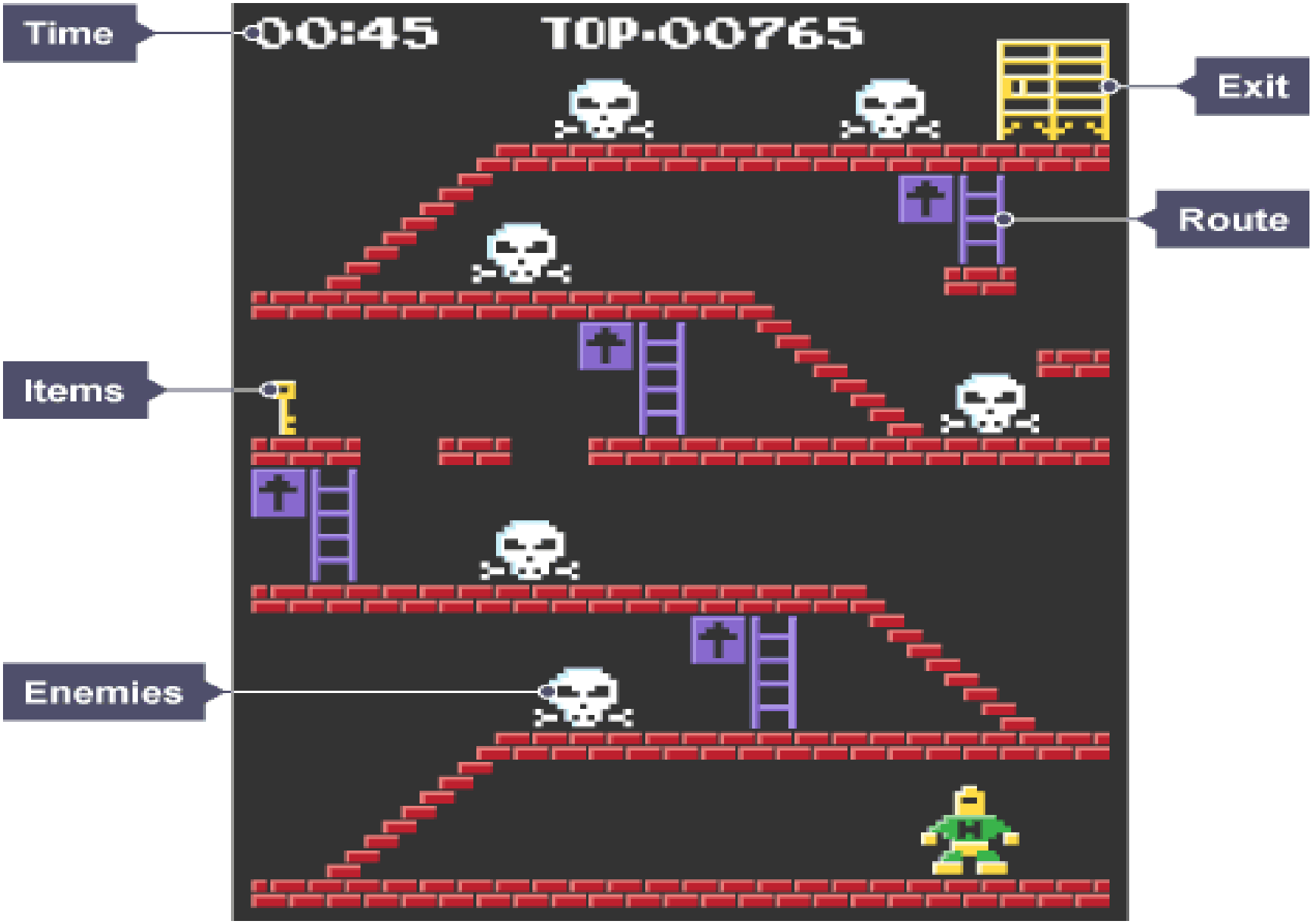
You need to decide what to do with your group of friends.

- If all of you like different things, you would need to decide:
  - what you could do
  - where you could go
  - who wants to do what
  - what you have previously done that has been a success in the past
  - how much money you have and the cost of any of the options
  - what the weather might be doing
  - how much time you have

- From this information, you and your friends could decide more easily where to go and what to do – in order to keep most of your friends happy.

- You could also use a computer to help you to collect and analyse the data to devise the best solution to the problem, both now and if it arose again in the future, if you wished.

Playing a videogame



Playing a videogame

- To complete a level you would need to know:
  - what items you need to collect,
  - how you can collect them, and
  - how long you have in which to collect them
  - where the exit is and the best route to reach it in the quickest time possible
  - what kinds of enemies there are and their weak points
- From these details you can work out a strategy for completing the level in the most efficient way.

- each complex problem was broken down into several small decisions and steps: **decomposition**
- only the relevant details were focused on: **abstraction**
- knowledge of previous similar problems was used: **pattern recognition**
- to work out a step by step plan of action: **algorithms**

- ✓ *If you were to create your own computer game, what are the types of questions you would need to think about and answer before you program your game?*

What has been described?

- Human based and Machine based computing
- Data and information
- Computational Thinking
- Corner Stones of CT

Credits

- *zendeux*
- *www.uaudio.com*
- *www.indiamart.com*
- *infosthetics.com*
- *www.mainstreamdata.com*
- *Google images*