INTRODUCING DIGITAL SKILLS FOR ALL INTO GET

Concept Document

Abstract

While our current world is shaped fundamentally by math and science, which we learn in school, our future world will be a digital world, where our lives are shaped by computers and connected devices.

Tink Tank.

C Labuscagne
Labuscagne.C@dbe.gov.za
Table of Contents

Background ................................................................................................................................................. 1
Context ........................................................................................................................................................ 1
Purpose ....................................................................................................................................................... 5
Areas of opportunity .................................................................................................................................. 5
  Stand-alone subject (preferred option) .................................................................................................. 6
  Stand-alone component, part of Technology in GET .............................................................................. 6
  Component of Life Orientation ............................................................................................................... 7
Description .................................................................................................................................................. 7
  Application Skills ...................................................................................................................................... 8
  Internet and e-Communication Skills ...................................................................................................... 9
  Data and Information Management Skills ............................................................................................... 9
  Computational Thinking (CT) Skills ........................................................................................................ 9
  Coding .................................................................................................................................................... 10
    Why Scratch? ..................................................................................................................................... 11
Outcomes ................................................................................................................................................... 12
Implementation Plan ................................................................................................................................14
  Curricula ................................................................................................................................................ 14
    Coding and Computational Thinking ................................................................................................. 14
    Other content .................................................................................................................................... 14
Resourcing ................................................................................................................................................. 15
  Teachers ................................................................................................................................................ 15
    Primary Schools (Grades 1 – 7): ......................................................................................................... 15
    Secondary Schools (Grades 8 – 9): ..................................................................................................... 15
  Software ................................................................................................................................................ 16
  Hardware ............................................................................................................................................... 16
  Extension Resources – Robotics ............................................................................................................. 17
  Internet Access ...................................................................................................................................... 17
  Learning and Teaching Support Material (LTSM) .................................................................................. 17
Annexure A ................................................................................................................................................... i
  Future jobs .............................................................................................................................................. i
  Jobs under threat ................................................................................................................................. ii
Annexure B – ScratchJr (Grades 1 – 3) ....................................................................................................... iii
  Interface ................................................................................................................................................ iii
  Activity Example .................................................................................................................................. iii
  Curricula ............................................................................................................................................... iv
  Assessments ......................................................................................................................................... v
Introducing *Digital Skills for All* into General Education and Training

**Background**

Programming, in the form of the subject Computer Studies Higher Grade (HG), in school was first introduced as an additional subject in the middle 70s at centres in the big cities to serve as an extended opportunity for gifted learners. Towards the middle 80s, Computer Studies HG was expanded to all high schools that opted to offer it as an additional subject after hours. The offering was similar to that of Additional Mathematics HG at the time and aimed at high flyers. Around the beginning of the 90s, Computer Studies was introduced as a full-time subject.

The fact that the subject was aimed as an extended opportunity for top achievers, resulted in very high standards with the result that it became inaccessible to average learners.

The subject was later extended to include a Standard Grade component that focused mostly on end-user applications, like the Computer Applications Technology (CAT) subject introduced as part of the National Curriculum Statement (NCS).

With the introduction of the NCS, Computer Studies HG was replaced by Information Technology (IT). The high standards were maintained and the subject was still out of reach for average learners, resulting in very low numbers taking the subject. With no grounding in the lower grades, implying a very steep learning curve, the numbers keep declining. Currently only 0.7% of Grade 12 learners take IT as a subject.

**Context**

Currently programming is only offered from Grade 10 – Grade 12 as part on the Information Technology (IT) curriculum.

According to the Communications of the Association of Computing Machinery (CACM), today many people want to know programming ...“to make themselves more marketable and effective in some other job...”

Today, computing is involved in almost all aspects of our lives and knowing how to code has numerous benefits to careers and beyond and it will make anyone more marketable and effective in any job.

Therefore, it is essential that learners need to learn to code from an early stage. Learning to code also develops higher order thinking skills, including computational thinking and problem solving, which would have the added potential advantage of improving achievement in all subjects.

The role of Information and Communications Technologies (ICTs) in everyday life, including the workplace, is becoming increasingly important. According to Forbes (2014), the technology sector is set to grow faster than all but five industries by 2020. The increasing dependence that companies have on computers could be contributing to the growth.

Per the Apollo Research Institute, healthcare and information technology are set to experience significant employment gains in the coming years.

Forbes (2015) suggests that the computer systems design and related services industry (programming, systems design and integration and facilities management) has been steady since 2010 and indicates a growth of nearly 18 percent during the last 12 months. The increasing dependence on computers might also be a reason why this industry has performed well for the last several years. “Even in 2009, when almost all industries had sales declines, computer systems design grew about 4 percent, which usually signifies a stable industry.”
Analysts also predict that monitoring and maintaining software and networks as well as analysing and recommending the best cyber security practices, are expected to grow at a rate of 36.5 percent between 2012 and 2022.

What people do with computing is changing, and growing daily. Programming is a medium, a literacy, a form of communication and expression. More and more people will use it. Computational thinking has become an essential skill and everyone should be taught computational thinking and coding.

The National Scarce Skills List: Top 100 Occupations in Demand (DHET, May 2014) identifies the following:

- Joint Initiative on Priority Skills Acquisition (JIPSA)
  - High-level, world-class engineering skills
  - Artisans and technical skills
  - Mathematics and Sciences teachers
  - ICT skills

- National Development Plan (NDP)
  - Critical shortages of doctors, engineers, IT professionals
  - Need to increase the number of health professionals (doctors and nurses), engineers, technologists, technicians and artisans, mathematics and science teachers

- New Growth Path (NGP)
  - Engineers: 30 000 additional engineers by 2014
  - Artisans: 50 000 additional artisans by 2015
  - ICT skills: DBE should ensure that ICT skills are taught in all secondary schools

- Strategic Integrated Projects (SIPs)
  - Engineering and built environment field:
    - Engineers, technicians, artisans, mathematics and science teachers, IT professionals

- HRDC report on Production of Professionals
  - Need for production of professionals in the following fields:
    - Engineering, mining, health care and the built environment

The following IT occupations appear in the Department of Higher Education and Training’s Scarce Skills List of May 2014:

<table>
<thead>
<tr>
<th>Position on list</th>
<th>Scarce Skills Occupation Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>ICT Systems Analyst</td>
</tr>
<tr>
<td>45</td>
<td>Software Developers (Programmers)</td>
</tr>
<tr>
<td>66</td>
<td>ICT Project Manager</td>
</tr>
<tr>
<td></td>
<td>Computer Networks &amp; Systems Engineer</td>
</tr>
<tr>
<td>79</td>
<td>Computer Network Technician</td>
</tr>
<tr>
<td>80</td>
<td>Network Analyst</td>
</tr>
<tr>
<td>90</td>
<td>Chief Information Officer</td>
</tr>
</tbody>
</table>

Table 1: DHET Scarce Skills List - May 2014

If one considers that programming is a compulsory subject for engineering studies, the list in the table above could be expanded to include all engineering courses.

For a list of future jobs and for a list of jobs under threat, refer to Annexure A.
An article in Memeburn (September 2014, https://memeburn.com/2014/09/south-africa-needs-get-kids-interested-in-computer-science-or-risk-falling-badly-behind/), states that South Africa needs to ensure that kids become interested in computer science, or risk falling badly behind. It also states: “Technology is an important part of today’s society, and incorporating it into the classroom equips students to better make the transition to the workplace. In addition, studies have shown that children conversant with technology show improvements in their writing, reading and maths skills”.

Coding alone will not fully equip learners for the 21st century (as 21st century skills are more than just coding) and the 4th industrial revolution. Therefore, learners need to be exposed to a broader range of ICT skills to support functioning in a changing world. This would include the use of ICTs and the understanding of data and information to make sense of the world. ICTs have become, within a very short time, one of the basic building blocks of modern society and the importance of digital skills is widely acknowledged.

Jason Gorman (CACM) claims that most people involved in computing are just ‘users’: “I believe that what’s needed is a much more rounded computing education for this majority, with IT blending seamlessly and ubiquitously into everyday lessons as well as home life.”

A significant impact on the middle-skill job market is in the humbler world of everyday software: spreadsheets and word processing and the ability to use software such as programs for medical billing and running computerized drill presses. To a large extent, a job seeker without the ability to use this software won’t even get in the door [Burning Glass Technologies: Digital Skills Gaps in the Workforce, March 2015]:

- Nearly eight in 10 middle-skill jobs require digital skills. Spreadsheet and word processing proficiencies have become a baseline requirement for the majority of middle-skill opportunities (78%).
- Digitally intensive middle-skill occupations are growing faster than other middle-skill jobs. Digitally intensive jobs have grown 2.5 times more rapidly than middle-skill jobs that do not require spreadsheets, word processing, or other digital skills (between 2003 and 2013, 4.7% growth for digitally intensive jobs compared to 1.9% growth for other positions).
- Digitally intensive middle-skill jobs pay more than middle-skill jobs that do not require a digital component. Digitally intensive middle-skill occupations offer 18% higher wages on average: $23.76 (±R325.91) per hour compared to $20.14 (±R276.26) per hour for all other middle-skill jobs.

Knowing how to answer your emails, to access a company’s files on Google Drive, or to tweak a website’s code are all digital skills that are increasingly sought after in today’s job market.

In the modern workplace, digital skills are highly valued; in the future, digital skills will be vital.

The term ‘digital skills’ refers to a range of different abilities, many of which are not only ‘skills’ per se, but a combination of behaviours, expertise, know-how, work habits, character traits, dispositions and critical understandings. These skills and competencies are interconnected and broadly complementary. They are also, in today’s technology-saturated communities, foundational to full participation in society.

---

1 21st Century skills (4Cs): critical thinking, communication, collaboration, creativity.
2 Will require computational thinking
3 Middle-skill jobs are defined as those occupations where fewer than 80% of the postings call for bachelor’s degree and hourly median wage is above the national living wage (US) and job postings do not call for digital intensive skills [such as coding]
and, as such, need to be developed and refined over time and according to the personal and professional circumstances of individuals (UNESCO, 2017).

Cornell University defines digital literacy as ‘the ability to find, evaluate, utilize, share, and create content using information technologies and the Internet’. By this definition, digital skills are any skills related to being digitally literate. Anything from the ability to find out your high-score on Minesweeper to coding a website counts as a digital skill.

Also, according to the UNESCO Working Group on Education: Digital skills for life and work, September 2017, recent analysis shows that, within developed economies, 90% of jobs require some level of digital skills, while more than one third of the labour force has an extremely limited ability to use ICTs productively. Digital skills are no less crucial in developing countries. In Kenya and India, for instance, an internet-connected device can serve as a bank, postal service, map, library, learning hub, translation utility and record keeper, all rolled into one.

Furthermore, a study done by Griesel and Parker (Graduate Attributes: A baseline study on South African graduates from the perspective of employers, HESA, 2009) about the attributes of South African graduates – from the perspective of employers, suggests that employers value the following from graduates:

- Communication skills (Written and presentation)
- Information skills
- Fluent (i.e. expertise and effective use) users of technology
- Understanding of technology
- Problem solving skills
- Enquiry and research skills

The foundation for all the above skills could all be taught in a well-designed Digital Skills for All curriculum.

As digital technologies now underpin effective participation in key areas of life and work, many countries are now understanding the importance of ICTs and the need to master the basic skills and concepts as part of basic education. For example, the United Kingdom’s (UK) new curriculum introduces computational thinking and coding concepts from the age of five years.

In a report, New Skills Now: Inclusion in the Digital Economy: Accenture (2017), ICT skills for a digital economy are classified as follows:

From the above, it is clear that coding and working with data and information, as well as understanding and using ICTs are important for full and effective participation in a workplace and advancing economy.
UNESCO further asks the following questions:

“What are the educational implications of the ‘broadband society’ for the development of digital skills for life and work?”

“How can education support the equitable development of digital skills for all? How can we advance digital skills development continually, in the context of fast-changing technologies?”

In response to the questions above, UNESCO suggests the following:

- Ensure digital literacy for all
- Teach coding skills to children and young people
- Foster ‘soft’ and ‘complementary’ digital skills, e.g. 21st century skills, responsible use of ICTs, etc.

Based on the context described above, the DBE proposes a Digital Skills for All curriculum to be introduced in GET that needs to

- enhance learner capacity for effective learning and improved achievements
- prepare learners for enhanced living and employment opportunities

Furthermore, a Digital Skills for All curriculum also needs to empower teachers.

**Purpose**

The purpose of a Digital Skills for All curriculum is to provide understanding and skills to

- function in a digital and information-driven world
- prepare for the 4th Industrial Revolution (4IR)
- develop computational thinking skills to support coding as well as all other subjects
- equip learners with digital and ICT skills and to transfer these skills to solve everyday problems
- develop a new generation of creative, systemic thinkers that can use coding to express their ideas

“For our current world is shaped fundamentally by mathematics and science, which we learn in school, our future world will be a digital world, where our lives are shaped by computers and connected devices.” (Tink Tank)

**Areas of opportunity**

There is no GET subject that provides for coding or computational thinking as a basis for coding or other digital skills required to function in a digital world.

The schooling system should not only focus on coding, but needs to include a broader range of skills to prepare learners for the demands of the 21st century, changing world and the 4th industrial revolution (digital revolution).

When considering coding at GET level, we should also consider computing as a pyramid:

At the base, we have computer users (using software such as word processing, spreadsheet, presentation, etc.), who will probably make up the biggest part of the pyramid.

People who code are towards the top of the pyramid.
At the top, are computer scientists – the serious programmers who advance the concepts, design the programming languages and “push the envelope”.

Therefore, the schooling system needs a subject, Digital Skills for All to be introduced at GET-level that will support computational thinking, critical thinking, problem solving, communication and collaboration as well as using ICTs effectively and coding.

It is suggested that a Digital Skills for All curriculum is introduced as an independent subject for one hour per as a non-examinable subject – assessment should be performance-based (e.g. portfolio with projects), i.e. focusing on product and process.

Research suggests that any offering of less than one hour per week, would not add value, therefore the subject could be introduced considering one or more of the following models:

Stand-alone subject (preferred option)
A subject in its own right, with dedicated time, taught by skilled teachers that specialise in Digital Skills.

Advantages:
• As a stand-alone subject, teachers could specialise in the subject resulting in maximum benefits for all learners, e.g. in schools offering CAT and IT, these teachers could be utilised (Gr 8 and 9).
• Additional, dedicated time will ensure that the curriculum is taught for maximum benefit.

Challenges:
• This option will require an additional one hour per week as a minimum (extending school time).

Stand-alone component, part of Technology in GET
Using one hour per week from the Technology subject in GET (leaving Technology with less time)

Advantages
• No additional time required (extension of existing teaching time)
• Could be taught by knowledgeable Digital Skills teachers, e.g. CAT and IT teachers and teachers could specialise in the subject resulting in maximum benefits for all learners.

Challenges
• This option will require the Technology subject’s time to be reduced by one hour per week to provide for an hour of Digital Skills for All. However, the current Technology curriculum is already overloaded and teachers may not be able to complete the curriculum with less time.
• In the Intermediate Phase, Technology is already combined with Natural Sciences and adding another component may make the subject too diverse which could pose a challenge in terms of teaching all these diverse skills.
Component of Life Orientation

Digital Skills for All could become a component of Life Orientation, which is compulsory for all learners.

Advantages

• Life Orientation is a compulsory subject for all learners in GET.

Disadvantages

• Removing some existing content, else increasing time allocation for Life Orientation to 3 hours per week (which would nullify the advantage above)
• Adding Digital Skills to the curriculum may make the subject too diverse which could pose a challenge to find teachers who could teach all these diverse skills

Considering all of the above, it is strongly suggested that

Description

When learning about computing you are thinking about thinking, about how thinking works.

"Everyone in this country should learn how to program a computer, because it teaches you how to think!" – Steve Jobs

A Digital Skills for All curriculum at school level should teach learners the understanding and skills:

• to function in a digital environment and to understand how ICTs and the Internet of Things (IoT) could work for them
• that enable the use of applications in an integrated fashion to solve everyday problems
• that enable learners and teachers to engage in e-learning and blended learning
• to articulate and think computationally, logically and creatively
• to use ICTs responsibly through the understanding of the legal, ethical, environmental, social, safety, security and health issues
• to use ICTs to find authentic and relevant information, process the data/information, make decisions and communicate findings in appropriate presentation media and identify correct curriculum-aligned content and distinguish it from misleading, false content or fake news
• that enable learners to apply 21st century skills across subjects and which will prepare them for the 4th industrial revolution

To address the above, the Digital Skills for All curriculum should include the following pillars:
Learners should not only master ICT skills, but also utilise ICTs to improve learning

**Application Skills**

- The efficient use of applications such as presentation software, word processing software, spreadsheet software and e-mail software as well as basic bitmap graphics editing and basic vector editing.
- Learning how to use these tools, understanding how and when to use these tools to achieve a purpose, such as completing a given project, identifying situations where each tool will be helpful, choosing the most appropriate tools for a particular task, and using these tools in combination to solve real-life problems.
• Technical: basic printer and installation of software; troubleshooting basic errors, e.g. mail not downloading on mobile phone

Internet and e-Communication Skills
• Learning how to use the Internet and other ICT tools, understanding how and when to use these tools to achieve a particular purpose, choosing the most appropriate tools for a particular task, and using these tools in combination to solve real-life problems.
• Apply a fundamental understanding of the ethical/legal issues surrounding the access and use of Internet and ICTs, especially in the context of social media, trolling, bullying, etc.
• Technical: Bluetooth/WIFI/connections; cost vs options

Data and Information Management Skills
• Use technology as a tool to learn, research, organise, evaluate and communicate information.
• Use digital technologies (computers, smart phones/tablets, media players, GPS, etc.), communication/networking tools and social networks appropriately to access, manage, integrate, evaluate and create information to successfully function in a digital world and a knowledge economy.

Computational Thinking (CT) Skills
Computational thinking is critical in the 21st Century and for the fourth industrial revolution, as it includes a collection of diverse skills to do with problem solving. People such as Wing (2008) calls for introducing computational thinking to young children as a formative skill together with literacy and numeracy.

“Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent.” [Brown] It is seen as to stand separately from problem-solving as it is more about capturing the preciseness of expression that is required to write programs and interact with precise systems.

Coding is a unique in the way it brings all these diverse skills together and it is one of the big advantages of teaching learners to code, as learning to program requires computational thinking skills. Before computers can be used to solve a problem, the problem itself and the ways in which it could be resolved must be understood.

It includes some obviously important skills that most subjects help develop, like creativity, the ability to explain and team work. It also consists of some very specific problem solving skills such as the ability to think logically, algorithmically and recursively.

CT is a process that generalises a solution to open ended problems. Open-ended problems encourage full, meaningful answers based on multiple variables, which require using decomposition, data representation, generalization, modeling, and algorithms found in CT.

There are four key techniques (cornerstones) to computational thinking:
• 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 and seeing problems as finite chunks which can be re-used rather than re-built over and over again.
• algorithms - developing a step-by-step solution to the problem, or the rules to follow to solve the problem
Lu and Fletcher (2009) suggest that the focus should not be on coding, but that computational thinking should be seen as a preparation for coding.

Computational thinking skills serve as foundation for coding and could be introduced using activities without computers, called unplugged activities. These activities are more than just for those days that computers are not functioning -

Such as activities freely available from Computer Science Unplugged (http://csunplugged.org/), code.org as well as Talent Search, to introduce learners to programming principles and underlying concepts in a manner that is separated from the distractions and technical details we usually see with computers.

Coding
Coding could be introduced using a free educational programming tool such as Scratch. Foundation phase learners could be introduced to coding through Scratch Junior whilst learners from Grade 4 – 9 could use Scratch 2.0. Refer to Annexure C and Annexure D for more information and examples of ScratchJr and Scratch 2.0.

With Scratch, learners can program interactive stories, games and animations — it affords learners the opportunity to develop the ability to code computer programs, which is an important part of literacy in today’s society. Scratch “… lets you create your own animations, games, and interactive art and enable the development of technological fluency” (Malan & Leitner, 2007).

When people learn to code in Scratch, they learn important strategies for solving problems, designing projects and communicating ideas. It also helps learners to learn to think creatively, reason systematically, and work collaboratively — essential skills for life in the 21st century.

Resnik et al. (2009) claim that, with Scratch, children learn important mathematical thinking and concepts of computational thinking as well as to think systematically and creatively. Also, as they develop projects with personal meaning, they become eager to learn more and develop concepts and thinking that are transferable to other subjects such as mathematics, technology and physical sciences.

Through Scratch programming, learners could also be engaged in robotics using Scratch Extensions that make it possible to connect Scratch projects with external hardware or sources of information on the web such as: https://wiki.scratch.mit.edu/wiki/Scratch_Extension#HTTP_Extensions

- **PicoBoard**, a piece of electronic equipment that allows Scratch to interact with the outside world. It can sense light and sound, along with having a button and slider, and alligator clips.
- **Lego WeDo**, a Robotics Kit that is a simple robotics tool designed for ages 7–11. It allows users to design their own robots, and then program the robots using Scratch.
- **Lego League**, a robotics program for 9 to 16 year olds, which is designed to get children excited about science and technology – and teach them valuable employment and life skills. Groups of learners engage in programming an autonomous robot (using the LEGO® MINDSTORMS® robot set) to score points on a thematic playing surface, creating an innovative solution to a problem as part of their project.
- **Hummingbird**, is a robot created for education in 2010 by BirdBrain Technologies. It is powered via a USB cable. It is easily connected to Scratch via BirdBrain Robot Server.
- **GoPiGo Raspberry Pi Robot**, can control a Raspberry Pi robot with Scratch Programming. Scratch can be used to control the robot motors, LEDs, and sensors.
• **GrovePi for the Raspberry Pi**, is an Internet of Things (IoT)\(^4\) device built on the Raspberry Pi. Scratch can be used to control LEDs, and read sensors like an ultrasonic sensor, noise sensor, and light sensor.

• **Joystick** (a lever that can be moved in several directions to control the movement of an image on a computer or similar display screen), with the *JoyTail* extension, Scratch can use remote sensor connections to connect to a joystick.

• **Webcam**, to use *Microsoft Kinect*-like features (form of Augmented Reality (AR)) in Scratch. The video feed can be shown on the stage (with transparency), and the motion of the video can be obtained as well.

• **Drones**, learn to program Drones easily with Scratch using a *Scratch Extension* developed by Camp K12 – see GitHub [https://github.com/campk12/ScratchForARDrone](https://github.com/campk12/ScratchForARDrone)

**Why Scratch?**

Scratch was created by Massachusetts Institute of Technology (MIT) and designed especially for ages 8 to 16.

Scratch has a huge community with numerous resources that includes learning guides, teaching guides, lessons, videos, etc.

Scratch offers a low floor (easy to get started, even with no experience of coding), wide walls (allows for all kinds of projects – from animations and games to science experiments) and high ceiling (the possibility to create complex projects) with a playful experience that teaches novices to code in an experimental fashion – it starts where the beginner is and allows for incremental development. It is therefore possible to use it throughout the curriculum, from Grade 1 to Grade 9.

Scratch is a pathway to high level programming languages as it allows novices to develop coding constructs and to focus on logical thinking before focusing on syntax, which is one of the features of high level programming languages (designed for industry) that makes them challenging.

Numerous research projects ([https://scratch.mit.edu/info/research/](https://scratch.mit.edu/info/research/)) suggest the value of Scratch to teach coding to novices, e.g. Meerbaum *et al.* (2010) found that a programming course that used Scratch, resulted in meaningful learning. Students were able to internalise programming concepts and showed improved cognitive achievement.

Furthermore, there is already a cohort of IT teachers that knows Scratch.

---

\(^4\) The internet of things (IoT) is a computing concept that describes the idea of everyday physical objects (e.g. household appliances) being connected to the internet and being able to identify themselves to other devices. The internet of things (IoT) is a computing concept that describes the idea of everyday physical objects being connected to the internet and being able to identify themselves to other devices.
Outcomes
The following diagram shows the outcomes of the *Digital Skills for All* curriculum for the different phases:
The Digital Skills for All curriculum as envisaged would also

- enhance learner capacity for effective learning and improved achievements
- prepare learners for enhanced living and employment opportunities
Furthermore, a *Digital Skills for All* curriculum has the potential to also empower teachers.

**Implementation Plan**

The following broad implementation plan is suggested:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Time frame</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop framework for Grade 8 &amp; 9</td>
<td>2018</td>
<td>DBE/Unisa/NWU</td>
</tr>
<tr>
<td>Develop Grade 8 &amp; 9 and Grade 1 curricula</td>
<td>2019</td>
<td>DBE/Unisa/NWU</td>
</tr>
<tr>
<td>Develop Lesson Plans and Activities (Grade 8, 9 and Grade 1)</td>
<td>2019</td>
<td>DBE/PEDs/Unisa/NWU</td>
</tr>
<tr>
<td>Teacher Training – Grade 8 &amp; 9 and Grade 1</td>
<td>2019</td>
<td>DBE/PEDs/Unisa/NWU</td>
</tr>
<tr>
<td>Pilot Grade 8 &amp; 9 in MST schools offering CAT and IT as well as Grade 1 in selected schools</td>
<td>2020</td>
<td>DBE/PEDs/Unisa/NWU</td>
</tr>
<tr>
<td>Revise Curricula for Grade 8 &amp; 9 and Grade 1 and Curriculum Development (Grades 2 – 7)</td>
<td>2020</td>
<td>DBE/Unisa/NWU</td>
</tr>
<tr>
<td>Develop Lesson Plans and Activities Development (Grades 2 – 7)</td>
<td>2020</td>
<td>DBE/Unisa/NWU</td>
</tr>
<tr>
<td>Teacher training Grade 8&amp;9 and Grade 1 &amp; Grade 2</td>
<td>2020</td>
<td>DBE/PEDs, Unisa/NWU</td>
</tr>
<tr>
<td>Implement revised curricula in Grade 1 and Pilot Grade 2</td>
<td>2021</td>
<td>DBE/PEDs/Unisa/NWU</td>
</tr>
<tr>
<td>Implement revised curricula in Grade 8 &amp; 9 in all CAT and IT schools</td>
<td>2021</td>
<td>DBE/PEDs/Unisa/NWU</td>
</tr>
<tr>
<td>Revise Grade 2 curriculum</td>
<td>2021</td>
<td></td>
</tr>
<tr>
<td>Teacher training Grade 1 – 3</td>
<td>2021</td>
<td></td>
</tr>
<tr>
<td>Implement revised curricula in Grade 2 and Pilot Grade 3</td>
<td>2023</td>
<td></td>
</tr>
<tr>
<td>Revise Curricula for Grade 2 and pilot Grade 3 Curriculum</td>
<td>2022</td>
<td>DBE/PEDs</td>
</tr>
<tr>
<td>Teacher training Grade 1 – 3</td>
<td>2021</td>
<td>DBE/PEDs</td>
</tr>
<tr>
<td>Implementation – all schools</td>
<td>2022</td>
<td>DBE/PEDs</td>
</tr>
</tbody>
</table>

Table 2: Suggested implementation plan for *Digital Skills for All* curriculum

---

**Curricula**

**Coding and Computational Thinking**

Curricula could be adapted for South African circumstances from existing, freely available curricula from various sources, e.g. Code.org®, and MIT Scratch

**Other content**

Requires development teams to develop curricula. The Computer Applications Technology (CAT) curriculum could serve as goal for development.
See Annexure B and Annexure C for Scratch curricula and Annexure E for curricula from Code.org.

**Resourcing**

The *Digital Skills for All* curriculum will require the following resources:

**Teachers**

*Primary Schools (Grades 1 – 7):*  
At least one dedicated teacher per school to teach the curriculum

*Secondary Schools (Grades 8 – 9):*  
Where schools are offering CAT and/or IT, these teachers could teach the curriculum, else at least one dedicated teacher per school would be required to teach the curriculum.

**Training**

- Teacher training for said teaching staff
- Training courses for existing CAT/IT teachers

Self-paced, open online courses are available for teachers (and learners) from edX.org ([https://www.edx.org/](https://www.edx.org/)) that strive to increase global access to quality education and connect learners to the best universities and institutions from around the world, e.g.

![edX Partners](https://example.com/edX_partners.png)

**How it works**

- Learn from the world’s best professors and leading industry experts through captivating lectures and presentations
- Build knowledge and expertise with interactive labs, experiments, and assessments
- Connect with learners from around the globe in easy-to-use discussion forums
- Take courses on your own schedule — anytime, anywhere

Examples of courses available from edX.org ([https://studio.code.org/courses?view=teacher](https://studio.code.org/courses?view=teacher)) that relates to *Digital Skills for All*

**Programming in Scratch:** [https://www.edx.org/course/programming-scratch-harveymuddx-cs002x-1](https://www.edx.org/course/programming-scratch-harveymuddx-cs002x-1)

**MyCS: Computer Science for Beginners (Scratch):** [https://www.edx.org/course/mycs-computer-science-beginners-harveymuddx-cs001x](https://www.edx.org/course/mycs-computer-science-beginners-harveymuddx-cs001x)


**Logic and Computational Thinking:** [https://www.edx.org/course/logic-computational-thinking-microsoft-dev262x-0](https://www.edx.org/course/logic-computational-thinking-microsoft-dev262x-0)

**Introduction to Computational Thinking and Data Science:** [https://www.edx.org/course/introduction-computational-thinking-data-mitx-6-00-2x-6](https://www.edx.org/course/introduction-computational-thinking-data-mitx-6-00-2x-6)

**Think. Create. Code.** [https://www.edx.org/course/think-create-code-adelaide-code101x-3](https://www.edx.org/course/think-create-code-adelaide-code101x-3)

The DBE should also collaborate with South African HEIs to develop self-paced MOOCs that are SACE accredited for this purpose and that will bear credits towards a qualification such as a certificate or diploma.
### Software

<table>
<thead>
<tr>
<th>Phase</th>
<th>Software required</th>
<th>Information</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grades 1 – 3</td>
<td><img src="https://www.scratchjr.org/about/info" alt="Scratch" /></td>
<td>Curricula, teaching, learning and assessment resources available: <a href="https://www.scratchjr.org/about/info">https://www.scratchjr.org/about/info</a></td>
<td>Free</td>
</tr>
<tr>
<td>Grades 4 – 9</td>
<td><img src="https://scratch.mit.edu/about" alt="Scratch" /></td>
<td>Curricula, teaching, learning and assessment resources available <a href="https://scratch.mit.edu/about">https://scratch.mit.edu/about</a></td>
<td>Free</td>
</tr>
<tr>
<td>Grades 1 – 7</td>
<td>Office suit &amp; e-mail software</td>
<td>Libre Office / Google docs / WPS Office</td>
<td>Free</td>
</tr>
<tr>
<td>Grades 8 – 9</td>
<td>MS Office / Libre Office</td>
<td>License fee required for MS Office (Need to prepare learners for CAT &amp; IT)</td>
<td>Free</td>
</tr>
<tr>
<td>Grades 1 – 9</td>
<td>Bitmap graphics editing software</td>
<td>GIMP <a href="https://www.gimp.org/">https://www.gimp.org/</a></td>
<td>Free</td>
</tr>
<tr>
<td>Grades 1 – 9</td>
<td>Vector graphics editing software</td>
<td>Inkscape <a href="http://www.inkscape.org/">http://www.inkscape.org/</a></td>
<td>Free</td>
</tr>
<tr>
<td>Grades 7 – 9</td>
<td>HTML editor</td>
<td>Notepad ++ <a href="https://notepad-plus-plus.org/">https://notepad-plus-plus.org/</a></td>
<td>Free</td>
</tr>
</tbody>
</table>

Also, an educational mobile application, the brainchild of a Nelson Mandela University student, will soon shape the next generation of software developers in the rural and township schools in the Eastern Cape. This app, designed by postgraduate student Byron Batteson, can teach pupils basic computer programming principles at primary school level and will be available for free [http://www.bizcommunity.com/Article/196/500/170411.html#more](http://www.bizcommunity.com/Article/196/500/170411.html#more).

Comment from Maryke Reed, IT teacher at Sterling, East London regarding the app:

> “Some of my Grade 9s had the opportunity to use this app at the ICT Summit in East London. None of them had any programming experience and I was amazed to see how they quickly and confidently managed to complete the levels making use of the little puzzle pieces. They were problem solving and had instant feedback whether their method was correct or not. They also started off with complicated solutions and soon realised how to solve the problem ‘smarter’. It is an addictive game as one always wants to solve the next level. It would be awesome if more levels could be added. This is an app that will definitely spark some excitement in programming and is perfect to explain the importance of.”

### Hardware

At Grades 1 – 7, Raspberry Pi 3 would suffice.

The Raspberry Pi has a small footprint, low power consumption, and low price. It is used to teach kids to code; put it in a robot for powerful processing capabilities; turn it in to a killer media center for your living room. This new Pi comes packed with new exciting fillings such as a 64 bit architecture and integrated 802.11n WiFi, Bluetooth 4.1 and BLE on the board.

The processor is a quad-core 64-bit 1.2GHz ARM Cortex A7 CPU which is 50% faster than its predecessor. Although RAM is the same as it’s predecessor, it is now faster at 900MHz. The same goes for its VideoCore IV graphics processor, which has jumped from 250MHz to 400MHz. The board’s footprint remains the same as the Pi2, but with more exciting features and backwards compatibility. It can be connected to any keyboard and any screen. These devices come preloaded with Scratch and Libre Office tools that will be required for the Digital Skills curriculum as well as other additional software.

A Raspberry Pi 3 board with case is around R650.00 each. Adding a keyboard would add another ±R250.00 and adding a display would add another ±R650.00.
A website with projects is available [https://projects.raspberry.org/en/](https://projects.raspberry.org/en/) and online training for teachers is also available [https://www.raspberry.org/training/](https://www.raspberry.org/training/)

**Extension Resources – Robotics**
Linking robotics with the coding will require additional resources such as PicoBoards, etc.

Depending on the extension, the cost could be between R1000.00 and R3000.00 + per kit.

**Internet Access**
All schools will require Internet access and data.

**Learning and Teaching Support Material (LTSM)**
Lesson Plans with activities need to be developed for all grades
Annexure A

**Future jobs**

The following jobs were identified as potential future jobs: (Singularity Summit)

<table>
<thead>
<tr>
<th>Job Description</th>
<th>Role/Industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D architects and large-scale 3D printer operators</td>
<td>Gene therapist / aesthetician (“designer genes”)</td>
</tr>
<tr>
<td>Artificial Intelligence (AI)-related work, such as AI trainer, robot ethicist</td>
<td>Hair stylist and other aestheticians</td>
</tr>
<tr>
<td>Android humaniser (Expressive Emotive developer)</td>
<td>Individualised curriculum designer / programmer (as opposed to a teacher)</td>
</tr>
<tr>
<td>Anti-terrorism/anti-hacking agent/investigator</td>
<td>Industrial designer</td>
</tr>
<tr>
<td>AR / VR content specialists / producers. More advanced gaming and visual arts designers.</td>
<td>Lifestyle manager for people who spend too much time in VR systems</td>
</tr>
<tr>
<td>Artist</td>
<td>Meat Printing Bio-Tech Specialist (meat as is currently farmed is unsustainable, and will likely be printed/manufactured. Several examples in existence now, were shown.)</td>
</tr>
<tr>
<td>Asteroid miner</td>
<td>Medical nanobot technician, nanobot programmer/designer</td>
</tr>
<tr>
<td>Avatar teleworker (this was in fact already in existence and demonstrated at the Summit) - a person’s face is shown on a robot and the robot moves according to the remote operator’s wishes, performing work at a distance (even over borders). A store in California is currently operated in this manner.</td>
<td>Neural implant interface designer / engineer. A current example was shown wherein the operator could move a robotic limb with their mind.</td>
</tr>
<tr>
<td>Big data analyst</td>
<td>Privacy consultant</td>
</tr>
<tr>
<td>Bioethicist</td>
<td>Psychologist</td>
</tr>
<tr>
<td>Biotechnologist</td>
<td>Quantum cryptographer</td>
</tr>
<tr>
<td>Bitcoin and cryptocurrency broker</td>
<td>Rejuvenation doctor (re-growing limbs, reversing ageing)</td>
</tr>
<tr>
<td>Carbon accountant</td>
<td>Research evaluator/Fact checker</td>
</tr>
<tr>
<td>Child carer</td>
<td>Robotics engineers</td>
</tr>
<tr>
<td>Cloner of deceased persons or pets, cloning agency</td>
<td>Social media manager</td>
</tr>
<tr>
<td><strong>Computer programmer</strong></td>
<td>Space pilot, space tour guides</td>
</tr>
<tr>
<td>Construction Managers/Robot construction supervisor</td>
<td>Sustainability officer</td>
</tr>
<tr>
<td>Crowdfunding specialist</td>
<td>Sustainable energy technicians and designers</td>
</tr>
<tr>
<td>Cryptocurrency investigator (e.g. against money laundering)</td>
<td>Universal Basic Income supplier and/or anti-fraud investigator</td>
</tr>
<tr>
<td>Cybernetics: e.g. prosthetic limbs/robo-skeletons</td>
<td>User experience manager</td>
</tr>
<tr>
<td>Mechanical and Electrical Engineers</td>
<td>Video/Virtual companionship, Entertainer for hire (e.g. dinner speaker)</td>
</tr>
<tr>
<td>Ethicist; philosopher</td>
<td>Xenobiologist</td>
</tr>
<tr>
<td>Gaming tutor and strategist (gaming for money, much like spectator sports)</td>
<td></td>
</tr>
</tbody>
</table>
**Jobs under threat**

The following jobs were identified as possibly under threat in the context of current future-oriented technology: (Singularity Summit)

<table>
<thead>
<tr>
<th>Job Category</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Practitioner / Diagnostician (medical doctor).</td>
<td>Existing automated AI is 95% accurate on medical diagnosis already.</td>
</tr>
<tr>
<td>Legal advisors, advisors of various kinds</td>
<td></td>
</tr>
<tr>
<td>Manual labourers of all types - to be replaced with machine operators.</td>
<td>An example was shown of a mining robot which would be operated by a miner, and another example of a robot which precision-eliminates weeds in crops; another example shown was a robot cleaning a kitchen and intelligently placing items in a dishwasher or dustbin.</td>
</tr>
<tr>
<td>Livestock farmers</td>
<td></td>
</tr>
<tr>
<td>Banks and bankers, accountants (but not auditors)</td>
<td></td>
</tr>
<tr>
<td>Builders and architects</td>
<td></td>
</tr>
<tr>
<td>Reporters (news media)</td>
<td></td>
</tr>
<tr>
<td>To a lesser extent, surgeons</td>
<td></td>
</tr>
<tr>
<td>Drivers, taxi operators</td>
<td></td>
</tr>
<tr>
<td>To a lesser extent, aircraft pilots</td>
<td></td>
</tr>
<tr>
<td>Store floor staff / shop assistants of all kinds.</td>
<td>Already some shops in Sweden are unmanned and allow customers to select and check out items themselves. The Summit presenters proposed that online orders / orders placed with an app, and delivered by drones, will soon be normal. These solutions already exist (e.g. amazon.com). Robot shop assistants were shown to already exist.</td>
</tr>
<tr>
<td>Teachers, lecturers, universities and formal education institutions.</td>
<td></td>
</tr>
</tbody>
</table>
Annexure B – ScratchJr (Grades 1 – 3)

Interface

https://www.scratchjr.org/learn/interface

Activity Example
Curricula

These curricula introduce powerful ideas from computer science that are not usually highlighted in early childhood education. We use the term ‘powerful idea’ to mean a concept that children can learn and that will serve their critical thinking and problem solving abilities beyond the lifetime of a specific classroom technology. While these curricula were designed as tools for teaching ScratchJr, the ultimate goal is to equip students with skills that can be applied outside of the app. Abilities like programming, expressing through technology, and user-centered design are just a few of the powerful ideas covered in our lessons.

Printable Block Images

You can print high quality images of the ScratchJr blocks for classroom instruction. Download PDF

Animated Genres Curriculum

This curriculum provides students with the opportunity to learn all of the concepts in ScratchJr and apply these concepts in their own personal creations. At the heart of the curriculum are three types of interactive projects that can be made with ScratchJr. Read more

Playground Games Curriculum

In the eight lessons of this curriculum, children learn how to use ScratchJr as they re-create familiar playground games. Read more

Reinforcing Literacy and Math Curriculum

These curricular modules describe ScratchJr projects that reinforce the Common Core Standards. They could serve as games programmed by older ScratchJr students to be played by younger ScratchJr students. Read more

https://www.scratchjr.org/teach/curricula
Assessments

ScratchJr Solve-Its
Assess students' understanding of the relationship between the programming blocks and their associated behaviors.

Have your students watch the videos in the playlist and record their answers on an answer sheet.

What you need:
- YouTube Video Playlist (above)
- Student answer sheet
- Block sheet

The videos aim to assess students using 4 different methods:

Fix the Program: Students see a line of code and see a character act out that program. Then, they will see the character do a different action, and they must correct the original code to reflect the new action.

Circle the Blocks: Students are shown a list of ScratchJr programming blocks, then watch a character perform a program full-screen without seeing the code for the program. Students then circle the blocks from the list on their answer sheet that they believe make up the program.

Match the Program: Students are given several different lines of code on their answer sheet, then watch a character perform a program full-screen without seeing the code for the program. They then choose the code on their answer sheet that they believe matches the program they saw.

Reverse Engineering: Students watch a character perform a program in full-screen without seeing the code for the program. While watching, they should think in their head which ScratchJr blocks would make the character move the way it did. Then, students cut out blocks from their block sheet and arrange them in the order that best matches what they saw.

https://www.scratchjr.org/teach/assessments/solveit
Annexure C – Scratch (Grades 4 – 9)

Interface

Scratch interface with code to calculate the price of a pizza based on the toppings chosen:
Example of Scratch Code

Figure 5. Sample Scratch script (from Pong-like paddle game) highlighting computational and mathematical concepts.
Curricula

CREATIVE COMPUTING

AN INTRODUCTORY COMPUTING CURRICULUM USING SCRATCH

download the guide

WHAT IS CREATIVE COMPUTING?

Creative computing is... creativity.

Computer science and computing-related fields have long been introduced to young people in a way that is disconnected from their interests and values — emphasizing technical skills and a creative mindset. Creative computing instead fosters the development of shared connections to computing, by drawing upon creativity, imagination, and innovation.

Creative computing is... empowerment.

Many young people with access to computers see them as consumers, rather than designers or creators. Creative computing emphasizes the knowledge, practice, and understanding that gives young people control over the tools that they use in their daily lives.

Creative computing is... computing.

Engaging in the creation of computational artifacts empowers young people for more than careers as computer scientists or programmers. It supports young people's development as computational thinkers — individuals who can engage in computational thinking, practices, and processes in all aspects of their lives, across disciplines and contexts.

Get Started!

Begin your creative computing experience by downloading the guide

The guide can be used in a variety of settings: classrooms, clubs, museums, libraries, and more with creativity of learners of all ages and beyond. No prior experience with computer programming is required; only a sense of adventure.

when this sprite clicked
say hello for 2 secs
forever
change color of effect by 20
move 10 steps
play drum 11 times
move 30 steps
# What is ScratchEd?

ScratchEd is an online community where Scratch educators can share stories, exchange resources, ask questions, and find people.

## Stories

### Resources

- Explore the Resources section to find activities, videos, lesson plans, handouts, assessments, presentations, and other Scratch resources.
- ScratchEd Resources »

### Discussions

- An introductory computing curriculum using Scratch, the Scratch curriculum guide includes lesson plans, handouts, and sample projects.
- Scratch Curriculum Guide »

### Members

- These one-page summaries describe some of the big ideas behind creating, learning, and programming with Scratch.
- Creating with Scratch »
- Learning with Scratch »
- Programming with Scratch »

### Events

- This quickstart video introduces Scratch with a wide range of examples, and a demo of Scratch's basic mechanisms.
- Intro to Scratch Video »

## New to Scratch?

### Where can I find more curricular resources?

- Explore the Resources section of ScratchEd to find activities, videos, lesson plans, handouts, assessments, presentations and other Scratch resources.
- ScratchEd Resources »

### How can I help my students get started with Scratch?

- These resources can help you and your students create your first Scratch project.
- Getting Started Guide »
- Getting Started Video »
- Scratch: Starter Projects »
- Scratch Cards »
- My First Scratch Project: January 2012 ScratchEd Webinar »

### What are the big ideas behind Scratch?

- These one-page summaries describe some of the big ideas behind creating, learning, and programming with Scratch.
- Creating with Scratch »
- Learning with Scratch »
- Programming with Scratch »

### What is Scratch good for?

- In this webinar recording, Minna Piirainen and Karen Brennan discuss four questions about Scratch: what is Scratch? What is good Scratch? and what is Scratch good for?
- Four Questions About Scratch: January 2011 Webinar »

### How do I make a project?

- These resources can help you and your students create your first Scratch project.
- Getting Started Guide »
- Getting Started Video »
- Scratch: Starter Projects »
- Scratch Cards »
- My First Scratch Project: January 2012 ScratchEd Webinar »

### What is Scratch?

- This quickstart video introduces Scratch with a wide range of examples, and a demo of Scratch's basic mechanisms.
- Intro to Scratch Video »

---

**http://scratched.gse.harvard.edu/resources/new-scratch**
Annexure D – Other computational thinking/coding curricula

Code.org
A non-profit organization dedicated to expanding access to computer science, especially increasing participation underrepresented groups

Other than the proprietary videos and artwork mentioned above, all curriculum and tutorial materials developed by Code.org are licensed to you for use under a Creative Commons Attribution-Non-Commercial-ShareAlike 4.0 Unported License. More information about this license can be found at: http://creativecommons.org/licenses/by-nc-sa/4.0/.

Code.org grants you a non-exclusive, transferable, non-sublicensable, limited right and license to access, view, use, and display the Code.org curriculum and tutorial materials. The Code.org curriculum and tutorial materials may only be used for non-commercial, computer science educational purposes. You may use these Code.org resources in a classroom where you charge students a fee to cover such costs as instructor compensation, venue, snacks, etc., so long as you do not represent the Code.org resources as your own creation or restrict access to the resources behind a paywall. If you are interested in licensing Code.org materials for commercial purposes, contact us.

Examples of curricula available from Code.org®

**Grade R – 5 (Computer Science Fundamentals)**

Computer Science Fundamentals was built with elementary school educators in mind. Courses A-F have been specifically tailored to learners in Foundation Phase to Grade 5, and no prior experience is assumed.

The lessons in CS Fundamentals are presented with the understanding that many teachers will not have any previous computer science training, and educators are therefore encouraged to learn along with their students.
Annexure E – Examples of unplugged computational thinking activities

Computer Olympiad

Elementary Level (Grade 5 and below)

**B3 Bracelet**

Emily has broken her favourite bracelet. The broken bracelet now looks like this:

![Broken bracelet](image)

**Question:**

Which of the following four bracelets shows what the bracelet looked like before it was broken? Write down the letter of your answer in the appropriate block on the answer sheet:

A: ![Alternative bracelet A](image)

B: ![Alternative bracelet B](image)

C: ![Alternative bracelet C](image)

D: ![Alternative bracelet D](image)
Junior Level (Grades 6 – 7)

C4 Tube System

A mouse is at the entrance of a tube system. It wants to reach the cheese at the end of tube 5.
The mouse always follows these commands:

1. Go down wards until a crossing
2. At the crossing, move through to the next vertical tube
3. Go to command 1

Question:
In which tube should the mouse start so that it will reach the cheese?

Write down the number of the tube in the appropriate block on your answer sheet.
Intermediate level (Grades 8 – 9)

B2 Blossom

Jane is playing a computer game.

First the computer secretly chooses colours for five buds without showing the colours. The available colours for each flower are blue, orange, and pink.

Jane has to guess which flower has which colour. She makes her first five guesses and presses the Blossom button to make the buds blossom. The buds, whose colours she guessed correctly, break into flowers. The others remain as buds.

Jane’s first go:

Jane then has another go at guessing and presses the Blossom button again.

Jane’s second go:

Question:
What colours did the computer choose for the flowers? Write down the letter of your answer in the appropriate block on your answer sheet:

A: blue pink blue orange orange

B: pink blue blue blue orange

C: pink blue blue pink orange

D: pink pink blue pink orange
Senior level (Grades 10 – 11)

C3 Rafting

Beavers build rafts. For river traffic control, all rafts should be registered. This means that each raft should have a licence plate with unique text. The text is made up of letters and digits as shown in the illustration below. The licence must start with the letter B and end with the digit 0 or 1.

![Image of raft with license plate SS010]

Question:
Which one of the licence plates cannot be registered? Write down the letter of your answer in the appropriate block on your answer sheet.

A: B60001
B: B60011
C: B60100
D: B00030
E: BSA001
F: BEO001
Elite level (Grade 12+)

C3 Kix Code

The Bebras Post Office uses postal codes that contain four characters. Unlike in South Africa each character can be a number or letter. To make the postal codes readable by machines, they convert the postal codes into Kix codes. In a Kix code, each character is represented by 4 vertical bars.

A code has 2 sections: upper and lower. The upper section contains only the middle and the top bars, while the lower section contains only the middle and the bottom bars.

This table shows the codes for several characters:

Example:
The Kix code for “G7YO” is

Question:
Another postal code has this Kix code. What is the postal code?

Write the postal code in the appropriate block on your answer sheet.
Examples of activities from **Computer Science Unplugged** ([http://csunplugged.org/](http://csunplugged.org/)) include the following concepts:

### Data: Representing Information
- Binary Numbers
- Image Representation
- Text Compression
- Error Detection
- Information Theory
- Sound Representation
- Databases

### Putting Computers to Work: Algorithms
- Searching Algorithms
- Sorting Algorithms
- Sorting Networks
- Minimal Spanning Trees
- Routing and Deadlock
- Phylogenetics
- Phylogenetics Unplugged
- Divide and Conquer
- Line drawing

### Telling Computers What to Do: Procedures
- Finite State Automata
- Programming Languages
- Class Simulation of a Computer - Unfinished
- Programming Languages
- Harold the Robot
- Information Hiding
- Cryptographic Protocols

### Sharing Secrets: Cryptography
- Treasure Hunt
- Programming Languages
- Marching Orders
- Sharing Secrets
- Public Key Encryption

**CS Unplugged** is a collection of free learning activities that teach Computer Science through engaging games and puzzles that use cards, string, crayons and lots of running around.

The activities introduce students to Computational Thinking through concepts such as binary numbers, algorithms and data compression, separated from the distractions and technical details of having to use computers. Importantly, no programming is required to engage with these ideas!

**CS Unplugged** is suitable for people of all ages, from elementary school to seniors, and from many countries and backgrounds. Unplugged has been used around the world for over twenty years, in classrooms, science centers, homes, and even for holiday events in a park!

The material is available free of charge, and is shared under a Creative Commons BY-NC-SA licence, which makes it easy to copy, adapt and share it.