Grade 8 Pre-Technical Studies

Term 2 - 43 complete lesson plans

Ready to teach, in the KICD lesson-plan format | 40-minute lessons | PDF and editable Word

ABOUT THIS PACK

43 complete, ready-to-teach Grade 8 Pre-Technical Studies lesson plans for Term 2, covering visual programming, composite materials, ceramics and cutting tools.

TOPICS IN THIS PACK

Topic	Name	Lessons
2.4	Visual programming	14
3.1	Composite Materials	7
3.2	Ceramics	7
4.1	Cutting Tools	15

INSIDE EVERY LESSON PLAN

- The specific learning outcomes, key inquiry question, core competencies, values and PCIs, and the resources needed.
- A timed lesson flow: what the teacher does and what the learners do, from introduction to summary.
- The key content (a worked example, notes or a practical with its expected results), then four practice questions with the answers for the teacher.
- The common mistake to watch for, and ideas for learners who need support and for those ready for more.
- A four-level CBC assessment rubric (EE, ME, AE, BE), homework, and a blank reflection space to fill in after teaching.

A TASTE OF LESSON 1

Programs and visual programming

By the end of the lesson, the learner should be able to: a) define programming and visual programming; b) explain how programs are used daily

> **Instructions for computers**
> A program is a set of instructions that tells a computer what to do. Programming (coding) is writing those instructions.
> Visual programming uses blocks or pictures that snap together instead of typed text, so beginners can code without spelling mistakes.

You also receive this pack as an editable Word file, so you can change any lesson to suit your class before you print it. Lessons follow the order of the KICD curriculum design.

CONTENTS

No.	Lesson	Topic	Page
1	Programs and visual programming	2.4 Visual programming	4
2	Types and examples of visual programming applications	2.4 Visual programming	7
3	The Scratch user environment	2.4 Visual programming	10
4	Features: input, processing, output and effects	2.4 Visual programming	13
5	Programming terms (part 1)	2.4 Visual programming	16
6	Programming terms (part 2)	2.4 Visual programming	19
7	Creating sequential instructions	2.4 Visual programming	22
8	Adding sound, animation and backgrounds	2.4 Visual programming	25
9	Repetition (loops)	2.4 Visual programming	28
10	Selection (if statements)	2.4 Visual programming	31
11	Variables: keeping score	2.4 Visual programming	34
12	Solving a problem: a quiz program	2.4 Visual programming	37
13	Testing and debugging	2.4 Visual programming	40
14	The value of visual programming	2.4 Visual programming	43
15	What are composite materials?	3.1 Composite Materials	46
16	Identifying composite materials in the locality	3.1 Composite Materials	49
17	Composition of concrete and bricks	3.1 Composite Materials	52
18	Composition of manufactured boards and coated paper	3.1 Composite Materials	55
19	Making papier-mache	3.1 Composite Materials	58
20	Uses of composite materials in the work environment	3.1 Composite Materials	61
21	Importance of composite materials	3.1 Composite Materials	64
22	What are ceramics?	3.2 Ceramics	67
23	Identifying ceramic materials in the locality	3.2 Ceramics	70
24	Properties: brittleness and hardness	3.2 Ceramics	73
25	Properties: heat and fire resistance	3.2 Ceramics	76
26	Properties: water and corrosion resistance	3.2 Ceramics	79
27	Uses of ceramics in the work environment	3.2 Ceramics	82
28	Importance of ceramics	3.2 Ceramics	85
29	What are cutting tools?	4.1 Cutting Tools	88
30	Saws: handsaws and hacksaws	4.1 Cutting Tools	91
31	Chisels and planes	4.1 Cutting Tools	94
32	Snips, scrapers and knives	4.1 Cutting Tools	97

33	Wire strippers and cutters	4.1 Cutting Tools	100
34	Selecting cutting tools for tasks	4.1 Cutting Tools	103
35	Safety when using cutting tools	4.1 Cutting Tools	106
36	Practical: sawing wood	4.1 Cutting Tools	109
37	Practical: using a chisel and a plane	4.1 Cutting Tools	112
38	Practical: hacksaw and tin snips	4.1 Cutting Tools	115
39	Practical: stripping and cutting wire	4.1 Cutting Tools	118
40	Caring for cutting tools	4.1 Cutting Tools	121
41	Practical: cleaning and storing workshop tools	4.1 Cutting Tools	124
42	Importance of cutting tools in the work environment	4.1 Cutting Tools	127
43	Review: cutting tools	4.1 Cutting Tools	130

Programs and visual programming

Strand 2.0 Communication > Sub-strand 2.4 Visual programming | Lesson 1 of 14 | 40 minutes

SCHOOL	TEACHER	DATE
CLASS / STREAM	TIME	ROLL (B / G)

SPECIFIC LEARNING OUTCOME(S)

By the end of the lesson, the learner should be able to: a) define programming and visual programming; b) explain how programs are used daily

KEY INQUIRY QUESTION(S)

- How are computer programs used in day-to-day life?

CORE COMPETENCIES, VALUES AND PCIS IN THIS LESSON

COMPETENCY	**Communication and collaboration**	Learners explain their thinking to a partner and report their findings to the group.
COMPETENCY	**Critical thinking and problem solving**	Learners choose a method, check their answers and give reasons for their choices.
VALUE	**Responsibility**	Learners look after shared materials and finish every task.
PCI	**Peer education**	Developed through the activities in this lesson.

LEARNING RESOURCES

Computers or tablets with Scratch (online or offline) | Printed block cards for unplugged lessons | Learners' exercise books

ORGANISATION OF LEARNING

Whole class for the introduction and the summary; pairs or groups of four to five for guided practice; individual work for the independent tasks.

KEY WORDS

program | programming | visual programming | coding block | sprite | stage | script | sequence | syntax | variable

THE LESSON AT A GLANCE

Introduction	Teacher models	Guided practice	On their own	Wrap-up
6 min	9 min	10 min	9 min	6 min

LESSON FLOW

Stage	Minutes	The teacher	The learners
Introduction	6	Greet the class and ask: "Key question: how are computer programs used in day-to-day life? What tells a mobile money app what to do?" Take two or three answers, link them to earlier work, then state today's outcome.	Answer the starter question, recall what they already know and read the outcome aloud.
Step 1: Teacher models	9	Take learners through "Instructions for computers" (key ideas on the next page), explaining as you go. Use computers or tablets with Scratch (online or offline), printed block cards for unplugged lessons, learners' exercise books. Ask learners to predict what comes next before you show it.	Watch, predict the next step, ask questions and note the key steps in their exercise books.
Step 2: Guided practice	10	Give questions 1 and 2. Move round the groups, ask "How do you know?" and correct errors kindly on the spot.	Work in pairs or small groups, explain each step to a partner and check one another's answers.
Step 3: Independent practice	9	Set questions 3 and 4 for individual work. Mark as you walk round and note who needs help in the next lesson.	Work on their own, then swap books with a partner to check.
Summary and conclusion	6	Ask two or three learners to state the key idea in their own words. Give this exit question and check the answers: "Why is visual programming good for beginners?"	Say the key idea in their own words and answer the exit question in the exercise book.

KEY IDEAS

Instructions for computers

A program is a set of instructions that tells a computer what to do. Programming (coding) is writing those instructions.

Visual programming uses blocks or pictures that snap together instead of typed text, so beginners can code without spelling mistakes.

Programs in daily life: mobile money, phone games, traffic lights, ATMs, washing machines, school timetables, weather apps.

PRACTICE QUESTIONS - WITH ANSWERS FOR THE TEACHER

No.	Question	Answer
1	What is a program?	**A set of instructions that tells a computer what to do.**
2	What is visual programming?	**Coding with blocks or pictures instead of typed text.**
3	Give three everyday uses of programs.	**Mobile money, games and traffic lights.**
4	Why is visual programming good for beginners?	**No typing errors; blocks snap together.**

WATCH OUT FOR

Some learners have little computer experience; pair them supportively.

ASSESSMENT RUBRIC - THE FOUR CBC LEVELS

Level	What the learner can do
Exceeds expectations (EE)	Can define programming and visual programming confidently and accurately, explain the method used, and apply the idea to new or unfamiliar problems.
Meets expectations (ME)	Can define programming and visual programming accurately in familiar situations; notices and corrects slips without help.
Approaches expectations (AE)	Can partly define programming and visual programming, or does so with prompts from the teacher or a partner; makes some errors.
Below expectations (BE)	Cannot yet define programming and visual programming without direct help; needs the idea taught again with concrete objects.

SUPPORT AND EXTENSION

Learners who need support	Act out a "robot" following instructions.
Learners ready for more	Write instructions for making tea as if for a robot. (Answer: Answers will vary)
Inclusion	Seat learners with low vision near the board and read aloud whatever you write; use large, clear, tactile materials; allow extra time; pair with a patient partner; use gestures or Kenyan Sign Language for the key words.

HOMEWORK AND FOLLOW-UP

Redo questions 3 and 4 neatly in the exercise book. Then tell someone at home what you learned about "programs and visual programming" and bring one question of your own to ask next lesson.

TEACHER SELF-CHECK

- Did every learner get a turn to handle materials or explain their thinking?
- Did most learners meet the outcome? Write down the names of those who did not.
- Was the pace right — did any step take too long or too short a time?
- Which competency and value did I see learners showing?
- What will I change the next time I teach this lesson?

REFLECTION ON THE LESSON / SELF-REMARKS

Write this after teaching - what worked, what to change, which learners need follow-up.

Types and examples of visual programming applications

Strand 2.0 Communication > Sub-strand 2.4 Visual programming | Lesson 2 of 14 | 40 minutes

SCHOOL	TEACHER	DATE
CLASS / STREAM	TIME	ROLL (B / G)

SPECIFIC LEARNING OUTCOME(S)

By the end of the lesson, the learner should be able to: a) identify types of visual programming applications; b) name examples

KEY INQUIRY QUESTION(S)

- How are computer programs used in day-to-day life?

CORE COMPETENCIES, VALUES AND PCIS IN THIS LESSON

COMPETENCY	**Communication and collaboration**	Learners explain their thinking to a partner and report their findings to the group.
COMPETENCY	**Critical thinking and problem solving**	Learners choose a method, check their answers and give reasons for their choices.
VALUE	**Responsibility**	Learners look after shared materials and finish every task.
PCI	**Peer education**	Developed through the activities in this lesson.

LEARNING RESOURCES

Computers or tablets with Scratch (online or offline) | Printed block cards for unplugged lessons | Learners' exercise books

ORGANISATION OF LEARNING

Whole class for the introduction and the summary; pairs or groups of four to five for guided practice; individual work for the independent tasks.

KEY WORDS

program | programming | visual programming | coding block | sprite | stage | script | sequence | syntax | variable

THE LESSON AT A GLANCE

Introduction	Teacher models	Guided practice	On their own	Wrap-up
6 min	9 min	10 min	9 min	6 min

LESSON FLOW

Stage	Minutes	The teacher	The learners
Introduction	6	Greet the class and ask: "Have you used an app to make a game or animation?" Take two or three answers, link them to earlier work, then state today's outcome.	Answer the starter question, recall what they already know and read the outcome aloud.
Step 1: Teacher models	9	Take learners through "Types and examples" (key ideas on the next page), explaining as you go. Use computers or tablets with Scratch (online or offline), printed block cards for unplugged lessons, learners' exercise books. Ask learners to predict what comes next before you show it.	Watch, predict the next step, ask questions and note the key steps in their exercise books.
Step 2: Guided practice	10	Give questions 1 and 2. Move round the groups, ask "How do you know?" and correct errors kindly on the spot.	Work in pairs or small groups, explain each step to a partner and check one another's answers.
Step 3: Independent practice	9	Set questions 3 and 4 for individual work. Mark as you walk round and note who needs help in the next lesson.	Work on their own, then swap books with a partner to check.
Summary and conclusion	6	Ask two or three learners to state the key idea in their own words. Give this exit question and check the answers: "What can MakeCode program?"	Say the key idea in their own words and answer the exit question in the exercise book.

KEY IDEAS

Types and examples

Educational: teaching coding and subjects (Scratch, Microsoft MakeCode, Sprite Box, Blockly-based apps).
Multimedia: making animations, stories and music.
Video games: designing and playing games.
Scratch (by MIT) lets you make stories, games and animations with sprites (characters) on a stage.
MakeCode programs devices such as small robots and microcontroller boards.

PRACTICE QUESTIONS - WITH ANSWERS FOR THE TEACHER

No.	Question	Answer
1	Name three types of visual programming applications.	**Educational, multimedia and video games.**
2	Name two visual programming applications.	**Scratch and Microsoft MakeCode.**
3	What is a sprite?	**A character or object in a Scratch project.**
4	What can MakeCode program?	**Devices such as robots and microcontroller boards.**