Northern Ireland Curriculum

3

Pupil Notes: Exploring Programming

- Look at the examples of programs your teacher has given you. Talk about the main purpose of each program and why it works well.
- With help from your teacher, make a storyboard plan for your program.
- Make a short list of what you need to test when the program is complete. Your teacher will help you with this.
- Use the programming software to make a new program file.
- With help from your teacher, source suitable subroutines and other assets to use in the program. Discuss how you will use these.
- Use the subroutines and assets you find to write your program.
- Save your final program using a filename that will help you and others know what it is.
- Use the list you made to test your program with a classmate. Talk about what works well and what you could improve.

Pupil Notes: Exploring Programming

- Look at examples of programs your teacher has given you and/or suggest some of your own. Talk about what the programs are for and why they work well.
- Read the scenario your teacher has given you and identify the problem.
- Make a storyboard plan for your program that takes into account its audience and purpose.
- Make a list of what you will need to test when the program is complete.
- Use the programming software to make a new program file.
- Source and edit suitable subroutines or code snippets and other assets to use in the program and discuss how you could combine and use these.
- Create your program to solve the problem.
- Save your final program using a filename that will help you and others to know what it is.
- Use the list you made to test your program with a classmate. Talk about what works well and what you could improve.

Northern Ireland Curriculum

5

Pupil Notes: Exploring Programming

- Find and review examples of programs, subroutines and algorithms relevant to your project. Identify the key characteristics that make them effective. Make a note of this research in your project diary.
- Read the scenario your teacher has given you and identify the problem.
- Make a plan for your program, using a combination of flowcharts, storyboarding or plain English instructions (pseudocode). Your plan should clearly explain how to solve the problem for your audience. Use programming language to support your planning.
- Think about the audience and purpose at all times while planning and editing your program. Remember to create a simple user guide (how to use the program) and include this in your end product.
- Create a list of aspects of the program that you will have to check or test when the program is complete.
- Make, find or edit subroutines or algorithms that you want to use in your final program, as your planning requires.
- Use your subroutines or algorithms to create a program to solve your problem, based on your pseudocode. Check your coding with any built-in error tracing function as you go along.
- Use the list you made to test and review your program with a classmate to ensure that everything works as you had intended. Make any changes that will improve your program.
- Save your program using a relevant filename and publish it in a suitable format that others will be able to use.
- Refer back to your research notes in your project diary. Make a note of at least two key characteristics that make your program effective for its intended audience and purpose.
- Consider which aspects or characteristics you would improve, if you were making the program again. Record your ideas in your project diary.
- Take a screenshot of your file, open in the software you used to create it (native file format). Place this at the end of your project diary.

Key Stage 3
Using ICT

Northern Ireland Curriculum

6

Pupil Notes: Exploring Programming

- Find and review examples of programs, subroutines and algorithms relevant to your project. Identify the key characteristics that make them effective. Consider why these characteristics are important and how you intend to incorporate them into your program. Make a note of this justification in your project diary.
- Read the scenario your teacher has given you and identify the problem.
- Consider the needs of your specific audience while planning your program. Note these in your project diary as objectives for your final program. Refer back to your notes during the coding process to ensure that you are meeting these requirements.
- Use flowcharts or plain English instructions (pseudocode) to create a detailed plan, taking into account the need for subroutines. Use programming language to support your planning.
- Think about the audience (end user) at all times while planning and developing your program.
- Remember to create a simple user guide (how to use the program) and include this in your end product.
- Develop a test plan that allows you to check your program methodically for bugs.
- Make, find or edit subroutines or algorithms that you want to use in your final program, as your planning requires. Add comments to your code to explain the function of each section and to justify any changes you have made to subroutines you have found.
- Create a program based on your pseudocode, using your subroutines or algorithms to solve your problem.
- Make sure the files you used to make your program are tidy. Delete any large files you no longer need. Save your work at key points and make a copy to avoid losing successful versions.
- Use your test plan and the software tools to test and review your program. Make a note in your project diary of any areas that you need to develop further. Try to fix any areas where the program does not perform as you intended (debugging). Repeat this review process until your finished product is as close to your original intention as possible, taking into account your objectives.
- Ask a user to test your completed program.
- Save your program using a relevant filename and export it in a suitable format that others will be able to use.
- Once you have published your program, make notes in your project diary to justify how your product meets your original objectives. Make specific reference to your audience and purpose. Record any objectives that it did not fully meet.
- Take a screenshot of your file, open in the software you used to create it (native file format). Place this at the end of your project diary.