

BTEC Nationals

IVA – LEARNER INSTRUCTIONS

Edexcel Level 3 BTEC National Certificate/Diploma

IT Practitioners (Software Development)

Unit 8: Programming Concepts and Practice

Unit 29: Software Development Project

Issued July 2004

For use during the remainder of the duration of operation
of the specification issued for May 2002



INSTRUCTIONS FOR LEARNERS COMPLETING IVAs

1. The Integrated Vocational Assignment (IVA) is a compulsory part of your qualification. If you do not complete the IVA you may not receive your certificate.
2. Your tutor(s) will tell you how long you have to complete the IVA and the access you may have to resources.
3. Read the IVA carefully and make sure that you understand the work you should hand in and what is required of you. If you are uncertain, discuss it with your tutor(s).
4. The IVA requires you to work by yourself and to produce original work. You should not share your work with any other learners. For example, if you produce an illustration or diagram electronically, you should not give it to another learner. Similarly, you should not accept and use such information from others. You are required to sign that the work submitted is your own.
5. If you work in a group at any stage, you must present your own responses to each task for assessment.
6. Information taken from sources for research, e.g. internet and textbooks, must be identified and not presented as your own work. You should list the sources used.
7. Some tasks may require Observation Records/Witness Statements. Your tutor(s) will organise for these to be completed and you must attach these to your submitted work.
8. In presenting your final work, you should not include draft work or reference materials such as handouts, notes and leaflets, unless the tasks specifically ask you to do so.
9. Presentation of your work:
 - Check that you have completed all tasks.
 - Label work with the appropriate task/sub task number.
 - Present tasks in the correct order.
 - Label each page with your name and page number.
 - Submit all electronic materials in paper format
 - Clearly label video or audio tapes submitted as part of your assignment.
 - All papers should be securely bound.
 - The completed IVA should NOT be presented in plastic envelopes, a box file or a lever arch file.

YOUR ASSIGNMENT TASKS

Unit 8: Programming Concepts and Practice

Scenario - School Games Competition

Every November, on the last Wednesday of the month, Salchester Primary School holds its annual school games competition. The school is very small and pupils from all classes belong to 'Houses' which compete for the House Cup. Each pupil belongs to either the Green or Yellow House. The names of pupils and numbers in each class are provided for you, together with their House.

The games that are played during the day are:

- Snap
- Cribbage
- Spillikins
- Junior Scrabble.

The structure of the competition is as follows:

- the Yellow and Green Houses compete against each other in a series of matches
- each match is 'best of three' single games of one type of game
- pupils cannot play other pupils from the same class
- pupils can only play another pupil once
- each pupil will take part in one match for each type of game.

To allow results to be entered, the aim of the program is to

- prepare a list of matches for the games competition
- identify the winners of each match on the schedule
- calculate the points awarded to each House.

This will allow the Head Teacher to present the House Cup, decorated with the right coloured ribbons, to the most successful House in the school at the end of the day. The program must be thoroughly tested before the games competition starts.

These are the specific requirements for your programme:

- the interface must use the House colours of yellow and green
- the interface must be very easy to use so that any of the staff on the day can deal with it
- there can be no slip ups on the day, so the programme must be thoroughly tested
- the list of matches must only be available to pupils on the day.

Data for School Games Competition Scenario

Pupils and their Houses (for year beginning Sept 05)

Class 1

Albert	Y	Bertha	G
Arnold	G	Betty	Y
Aswan	Y	Bella	G

Class 2

Charles	G	Denise	G
Colin	Y	Debra	Y

Class 3

Edward	G	Felicity	Y
Elias	Y	Freda	G
Earl	G	Fiona	Y

Class 4

George	G	Harriet	G
Gilbert	Y	Hebe	Y
Gerry	G	Helen	G
Gwyn	Y	Hilary	Y

Class 5

Ian	G	Jane	Y
Idris	Y	Jenny	G
Isaac	G	Jasmine	Y

Class 6

Keith	G	Laura	Y
Kenny	Y	Leila	G
Kevin	G	Linda	Y

Total in Green House = 18

Total in Yellow House = 18

Total Pupils = 36

Task 1

Investigate fully the problem described in the scenario. Decide how you would solve the problem, paying particular attention to your own requirements as well as those of the client. Determine the minimum and optimal system requirements.

a) You should cover:

- Your requirements
- The client's requirements
- The system requirements including:
 - User interface
 - Inputs
 - Outputs
 - Processing
 - Data storage

This task provides evidence for unit 8 P1

Task 2

Describe the alternative design approaches that could be used to create the program. You should include:

- a) A comparison of different design methodologies/approaches including RAD (Rapid Application Development), top down, bottom up, data driven, event driven.
- b) The different data structures that could be used to store and manipulate the data in your proposed system, showing examples for each appropriate type of structure. Identify the structures you propose to use in your system.

This task provides evidence for unit 8 M1

Task 3

- a) Apply top-down modular design techniques to produce the design documentation for the program, with enough detail to allow another programmer to implement your proposed system, based closely on your specified system requirements.
- b) Develop a test plan, as part of the design, which tests normal, erroneous and extreme data and includes tests for algorithms and interfaces.
- c) Your design documentation should cover the complete system and include, as part of the use of your chosen methodology/approach:
 - Structure diagrams, or recognized alternatives, showing sequence, selection and iteration
 - Data structures
 - Data storage
 - Inputs
 - Outputs
 - Processes
 - User interface
 - User initiated events
 - Data Dictionary including:
 - Procedures
 - Functions
 - Variables (Names, Data types, Data scope)
 - Modularisation (keeping module re-use in mind).

This task provides evidence for unit 8 P2, M2, P7, P4, P5, P3

Task 4

- a) Implement your design, including a suitable user interface.
- b) implement your test plan, documenting the tests and recording actions needed to resolve test failures.

This task provides evidence for unit 8 P6, P8, P9, P10

Task 5

Analyse your system, and evaluate:

- a) the effectiveness of the programming environment used
- b) the programme and testing procedures that were adopted.

This task provides evidence for unit 8 D1, D2

Part B: Unit 29 Software Development Project

You work for a software company and you are asked to produce a complete computer application using high level language programming tools. The stages involved will follow the systems development life cycle. For this assignment, it will not be possible to assess the maintenance stage, but any well developed program should always be easy to maintain provided that the technical documentation delivered with the product is full and accurate.

Your project must be for a real world situation, although this may be a simulation. It can be for any application, for example in business, education or industry.

Throughout the duration of the activity, you will be expected to calculate the time you spend on the project so that this could be charged to the client. You will be expected to prepare a plan using recognized planning techniques, which identifies tasks, resources, milestones and costs.

You will need to analyse, design, implement, test and evaluate your application. You will provide documentation appropriate and relevant to each stage including full technical and user material. The technical documentation will be such that another programmer is able to create and maintain your application without your support and guidance. The user guide will be aimed at all prospective users. Some help facilities may be built into the system you develop.

Finally, you will evaluate your completed system against the original specification and plan, and make recommendations for future developments.

Task 1

- a) Identify and describe the problem.
- b) Consider alternative designs which may fulfil the solution. Select an appropriate design. Document:
 - Designs you have considered
 - Reasons for selection of final design.
- c) Compare possible alternate solutions to the problem (these may be both IT and clerical solutions). You should select the most appropriate solution for your project.
- d) Consider the advantages and disadvantages of the different software that could be used to produce an appropriate solution. You should discuss the possible application software and programming languages, and identify the most suitable for your project.

This task provides evidence for unit 29 P2, P4, M1, M2

Task 2

Finalise your project proposal by producing a project plan. The plan should use recognised planning techniques and software.

- a) The plan should show clearly:
- All tasks to be performed.
 - All relevant deliverables and associated milestones.

The major milestones will come from examination of the tasks and system requirements.

- b) The plan should also show resource allocation and budgeting requirements. The costings of hardware, software and your time as developer will be involved here.
- c) This plan should be reviewed and updated whenever required as the differences will need to be discussed in the evaluation. If no alterations are required because you keep to time throughout, then this should also be recorded.

Your project records should include the original plan and updated / reviewed plans.

This task provides evidence for unit 29 P1

Task 3

Produce user and technical specifications for your project. You should use a recognised methodology to present this analysis; it will form part of the technical documentation. You should cover:

- a) The client's requirements including:
- Users
 - Objectives.
- b) The system requirements including:
- User interface (bearing in mind the aspect of colour blindness)
 - Inputs
 - Outputs
 - Processing
 - Data storage
 - Design constraints.

This task provides evidence for unit 29 P3

Task 4

- a) Produce the design documentation (with enough detail to allow another programmer to implement your proposed system, based closely on your specified system requirements) for the creation and maintenance of the application. Note any design problems for use in Task 7.
- b) Your design documentation should include:
- Structure diagrams or recognised alternatives which show sequence, selection and iteration
 - Data structures and storage
 - Inputs
 - Outputs
 - Processes
 - User interface
 - User initiated events using an action chart (if appropriate)
 - Data Dictionary
 - Test plan, to meet system and user requirements showing:
 - o Priority of test
 - o Test data to be used
 - o Reason for use
 - o Expected result.

This task provides evidence for unit 29 P5

Task 5

Implement the prototype following your design proposal carefully.

- a) Your developed prototype system must make correct use of the chosen language.
- b) The documentation should allow another programmer to maintain the program developed with ease. The completed documentation should include a program listing showing:
- Thorough code annotation – this annotation should also include relevant and appropriate variable names, explanations (not repetition) of the developed code and indentation of the code
 - Program code that matches the design structure (or equivalent) diagrams wherever possible.
- c) Any deviation from the design must be noted as it will form part of the evaluation in Task 7.
- d) Check that documentation is complete and covers all stages of the project.

This task provides evidence for unit 29 P6, P7, D1

Task 6

Use the test data you have formulated in Task 4 to test your program thoroughly. You must ensure that you note all aspects of the testing so that any functionality issues can be discussed in Task 7.

The evidence of testing may take a variety of forms including screen shots and witness statements. You should maintain a test log showing:

- Evidence of a range of appropriate tests being carried out
- Test plan from Task 4 being implemented
- For results of any tests that failed, action taken to fix the problem and evidence of re-running test.

This task provides evidence for unit 29 P8, M3, M4

Task 7

Produce a summary report for your project:

- a) You should evaluate every aspect of your application identifying strengths, weaknesses and scope for improvement. You should cover:
 - Design
 - Implementation
 - Testing
 - Meeting user requirements and user feedback
 - Programming environment.
- b) You should also provide an evaluation of your time management and your ability to follow the project plan. Decide whether your initial costings were accurate, and discuss any discrepancies.
- c) The application developed may yet be incomplete or have outstanding problems. Discuss any improvements you need to make. Identify any enhancements you would like to make to the system or any further user requirements.

This task provides evidence for unit 29 D2, D3

ASSESSMENT CRITERIA

For ease of reference, the assessment criteria from the unit specifications that are relevant to assessing and grading this IVA are repeated below. You should refer to the full unit specification for information on unit content.

Assessment Evidence Unit

Unit 8: Programming Concepts and Practice		
To achieve a pass grade the evidence must show that the Learner is able to:	To achieve a merit grade the evidence must show that the Learner is able to:	To achieve a distinction grade the evidence must show that the Learner is able to:
<ul style="list-style-type: none"> • analyse system requirements • apply top-down design techniques • produce a modular design • interpret and produce structure diagrams for sequence, selection and iteration • produce a data dictionary • make use of available facilities to produce a user interface • design a complete system • implement a test plan using designed test data • document tests and state the actions needed to resolve test failures • demonstrate a variety of testing procedures. 	<ul style="list-style-type: none"> • describe alternative design approaches including rapid application development, bottom-up design and data driven design • demonstrate appropriate and adequate design before implementation of a programme. 	<ul style="list-style-type: none"> • analyse the effectiveness of the programming environment used for the solution to a given problem • evaluate the programme and the testing procedures that were adopted.

ASSESSMENT CRITERIA

For ease of reference, the assessment criteria from the unit specifications that are relevant to assessing and grading this IVA are repeated below. You should refer to the full unit specification for information on unit content.

Assessment Evidence Unit

Unit 29: Software Development Project		
To achieve a pass grade the evidence must show that the Learner is able to:	To achieve a merit grade the evidence must show that the Learner is able to:	To achieve a distinction grade the evidence must show that the Learner is able to:
<ul style="list-style-type: none"> • produce a basic project plan • identify a suitable problem for a project • produce a simple user and technical specification • select an appropriate design • use the selected approach to meet agreed specifications • identify and implement software as appropriate • implement a solution • test the system. 	<ul style="list-style-type: none"> • compare a range of solutions and select the most appropriate solution • compare the various languages that could be used for the implementation of the project • demonstrate that requirements have been met • demonstrate thorough testing of the project. 	<ul style="list-style-type: none"> • provide full documentation for all stages of the project • evaluate the programme according to the original design specification and plan • propose recommendations for the future development of the project.