

Internal Assessment Resource

Achievement Standard Digital Technologies 91372: Construct a plan for an advanced computer programme for a specified task

Resource reference: Digital Technologies 2.45 v2

Resource title: Plan for Ruakaka Pizza Phone Ordering Programme

Credits: 3

Achievement	Achievement with Merit	Achievement with Excellence
Construct a plan for an advanced computer program for a specified task.	Skilfully construct a plan for an advanced computer program for a specified task.	Efficiently construct a plan for an advanced computer program for a specified task.

Student instructions

Introduction

This assessment activity requires you to construct a plan for an advanced computer programme that will perform certain specified functions for a pizza delivery business. Your teacher will specify the manner in which you plan the programme.

You will be assessed on how skillfully you plan the programme and the efficiency of the plan.

The criterion for excellence relates to the efficiency of the constructed plan for an advanced computer programme. All of the evidence for efficiency can be found within the plan e.g. reusing modules to reduce redundancy, variables and constants that are used effectively, comprehensive planning of programme and testing procedures.

This is an individual assessment activity. You will have 2 weeks of in-class time and out-of-class time to complete it.

Due Date: 11th May 2018

Task

Ruakaka Pizzas want to computerise their phone orders. Specifically, they want to be able to enter customer details, pizza(s) ordered and pick-up or delivery requirements into a computer and have it display the delivery details, itemised order, and total cost. Phone orders generally consist of several kinds of pizza.

You must construct a plan to allow your programme to meet the following specifications:

- The programme contains options for the phone operator to specify whether the pizza order is for pickup or delivery.
- If the order is for delivery:
 - the programme should collect the customer's name, address and phone number
 - a \$3 delivery charge should be added to the total cost.
- If the order is for pick up:
 - The phone operator should be asked to enter the customer's name into the programme.
- The programme should allow the phone operator to input how many pizzas the customer would like (maximum 5).
- A numbered menu of at least 12 pizza names should be presented to the phone operator. This menu should be stored in an indexed data structure (e.g. an array or list) and may be hard-coded within the programme so that pizza names don't need to be typed in every time the programme is run.
- Each pizza to be ordered should be selected from the choices available on the menu and the order information should be stored.
- The cost of the first seven (regular) pizzas on the menu is \$8.50 and the rest are \$5 more as they are gourmet pizzas.
- When the order is finished:
 - the names of ordered pizzas and their individual price should be displayed
 - the total cost of the order including any delivery charge should be displayed
 - customer name should be displayed
 - if the pizza is for delivery the address and phone number should be displayed.
- The programme should allow the operator to cancel the order.
- After the order information has been displayed the programme should be ready to accept another order or exit.

Note: There does not need to be any facility for editing the ordered list of pizzas. If the order is not right, the operator would just cancel it and start again from the beginning.

Your plan:

- should communicate the structure and function of a computer programme
- should be developed in the context of a target programming language
- may be communicated using a range of tools
- should not be expressed in programme code.

You can plan your program in natural language (as long as it is unambiguous and precise), diagrams (including screen layouts), or pseudocode.

Ensure that your plan specifies:

- variables, scopes and their data types
- indexed data structure
- a modular structure including procedural details of the modules
- expected and boundary input test cases.

Your finished plan must include at least one indexed data structure and be decomposed into user-defined modules with scoped variables, constants and derived values effectively so as to maximise the flexibility and robustness of the plan. Your chosen scope should match the way the variable is used.

Annotate your plan and use explanatory variable/module names so that the purpose of each part of the plan is clear. This includes documenting the plan with annotations on the function of the modules.

Include in your plan a testing schedule with input cases that ensures that your programme will work correctly on all inputs – expected, boundary, and exceptional.

To achieve with Merit you will need to demonstrate independence when constructing your plan

To demonstrate an efficient plan, modules will have clear and well-defined purposes. Interaction between modules will be minimised and there will be no duplication. The procedure structure of each module will be efficient.

Hand in your final plan.