

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Recycling Machine** | | | | |
| **Year** | **No.** | **Question** | **Marks** | **Answers** |
| 18/19  (SEM2) | 5 | Consider a software system that controls a recycling machine for returnable bottles, cans and crates. A customer can return all three types of item on the same occasion. The system must check, for each item, what type has been returned. The system will register how many items each customer returns and when the customer asks for a receipt, the system will print out what was deposited, the value of the returned items and the total return sum that will be paid to the customer.  This system has been extensively used as an example during the lecture and the practical sessions. A draft class diagram is provided in Figure 1.    Figure 1 Draft class diagram for Recycling Machine Implementation  Provide detailed answers to the following questions [10 marks each]:     1. Object-Oriented system development is about the role and responsibilities of individual classes and objects. Within the context of object-oriented design, what is the specific role or responsibility of the ‘Receipt Printer’ class? 2. Deposit item is an abstract type. Explain the term ‘abstract type’ with reference to the Deposit item. | 20 | From Lec Slides >>> **The system “interfaces” with the physical object “printer”, so we add an interface concept**  Q5a:  In the Object-Oriented development model, each classes and objects have their own role and responsibility. When the customer returned the items, the receipt printer will receive item, value and the sum of values from the Deposit item and Receipt basis. Then the receipt printer will print the receipt. Therefore, the main role of the receipt printer is to print the receipt.  Q5b:  Abstract types mean the types without instances or without definition and the implementation is completed in another class, and Normal classes cannot have abstract methods. In this scenario (figure 1) Deposit item have two variables: number and value. The sub classes of it: can, bottle and crate classes will override the methods in parent class (Deposit item). Thus, by making this abstract we force all child classes to implement the methods in the parent class. |
| 18/19  (Main) | 5 | Consider a software system that controls a recycling machine for returnable bottles, cans and crates. A customer can return all three types of item on the same occasion. The system must check, for each item, what type has been returned. The system will register how many items each customer returns and when the customer asks for a receipt, the system will print out what was deposited, the value of the returned items and the total return sum that will be paid to the customer.  This system has been given to you as example during the lecture and the first week practical sessions. A draft class diagram is provided in Figure 1.  Provide detailed answers to the following questions [10 marks each]:    a) The Customer Panel is the “interface” of the system. With direct reference to this example discuss the term ‘separation of concerns’, which is a major design principle for software systems.  b) What is the specific role or responsibility of the ‘Deposit Item Receiver class? | 20 | 1. In computer science, separation of concerns (SoC) is a design principle for separating a computer program into distinct sections such that each section addresses a separate concern. A concern is a set of information that affects the code of a computer program. A concern can be as general as "the details of the hardware for an application", or as specific as "the name of which class to instantiate". A program that embodies SoC well is called a modular. program. Modularity, and hence separation of concerns, is achieved by encapsulating information inside a section of code that has a well-defined interface. Encapsulation is a means of information hiding. Layered designs in information systems are another embodiment of separation of concerns (e.g., presentation layer, business logic layer, data access layer, persistence layer)   <https://simplicable.com/new/separation-of-concerns>   1. The role of the deposit item receiver is to receive the deposit items, the items are then classified and stored. It also acts as a communicator to the Receipt Basis. The Receipt Basis will be created when needed for the fist time. |
| 17/18 | 3 | Consider a software system that controls a recycling machine for returnable bottles, cans and crates. A customer can return all three types of item on the same occasion. The system must check, for each item, what type has been returned. The system will register how many items each customer returns and when the customer asks for a receipt, the system will print out what was deposited, the value of the returned items and the total return sum that will be paid to the customer.  This system has been extensively used as an example during the lecture and the practical sessions. A draft class diagram is provided in Figure 1.    Figure 1 Draft class diagram for Recycling Machine Implementation  Provide detailed answers to the following questions [10 marks each]:    a) What is the role or responsibility of the ‘Customer panel’ class?  b) The Receipt basis is the ‘database’ of the system. It could be implemented in many ways. With direct reference to this example discuss the term ‘encapsulation’. | 20 | 1. Customer panel initiates the classification and receipt printing action in the Deposit item receiver, initiating action in other objects. 2. Encapsulation is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects. |
| 17/18 | 3 | Consider a software system that controls a recycling machine for returnable bottles, cans and crates. A customer can return all three types of item on the same occasion. The system must check, for each item, what type has been returned. The system will register how many items each customer returns and when the customer asks for a receipt, the system will print out what was deposited, the value of the returned items and the total return sum that will be paid to the customer.  This system has been extensively used as an example during the lecture and the practical sessions. A draft class diagram is provided in Figure 1.    Figure 1 Draft class diagram for Recycling Machine Implementation  Provide detailed answers to the following questions [10 marks each]:     1. What is the role or responsibility of the ‘Receipt Basis’ class? 2. The Customer Panel is the “interface” of the system. With direct reference to this example discuss the term ‘separation of concerns’, which is a major design principle for software systems. | 20 | <https://en.wikipedia.org/wiki/Separation_of_concerns>  <https://warwick.ac.uk/fac/sci/physics/research/condensedmatt/imr_cdt/students/david_goodwin/teaching/modelling/l6_classdia.pdf> |