Object Oriented Programming

**Museum Exhibitions**

*National Museum* is organizing a series of special exhibitions and wants to allow its visitors to purchase tickets in the foyer so as to reduce queues at the ticket office. The exhibitions are:

* Roman Exhibition
* Warhol Art Exhibition
* World War 2 Exhibition

Each exhibition is open from 9am – 5pm but there are a limited number of tickets available for each day.

Tickets are categorized as follows:

* Adult Ticket - £10
* Child Ticket - £5
* Student Ticket - £7
* OAP Ticket - £8

When a customer wants to visit a particular exhibition, a ticket can be purchased at a ticket kiosk.

At the kiosk, the visitor chooses an exhibition and selects how many of each ticket type is to be purchased and, if space is available, is then told the total ticket price. The user can pay with a card or by cash.

If a card option is chosen, the user simply enters their card and the money is taken and the ticket(s) are produced.

If the cash option is chosen, then as money is entered, a message is displayed stating how much more money is still required. When the required money is entered the ticket(s) are produced. If any change is due, a message is displayed and the change is returned.

If the ticket kiosk has run out of a particular ticket, a message is displayed before any money has been given. The user will then have to buy the ticket at the counter.

A Java program is required to make the machine operate correctly. As the ticket machine has three main components – dealing with **Money,** dealing withseveral **Exhibitions** and dealing with several types of **Ticket**, the program will require classes to implement these components.

For Exhibitions, the requirements are:

**Properties:**

* A variable to store the exhibition name
* A variable to store the amount of daily tickets available for that particular exhibition

**Methods**:

* Suitable constructor(s)
* Suitable get and set methods
* Reduce the number of spaces available when tickets are sold and let the user know if the booking was possible

For Money, the requirements are:

**Properties:**

* A variable to store the amount of cash in the kiosk

**Methods:**

* Suitable constructor(s)
* Suitable get and set methods
* Receive the money deposited by the customer and use it to update the amount of cash in the kiosk.
* Receive the value of the change needed for the user and use it to update the amount of cash in the kiosk.

For Tickets, the requirements are:

**Properties:**

* A variable to store the name of the ticket type
* A variable to store the cost of the ticket
* A variable to store the number of the tickets in the kiosk

**Methods to**

* Suitable constructor(s)
* Suitable get and set methods
* Reduce the number of tickets when tickets are sold

The **application program** should allow the user to interact with the ticket machine so that a customer can book and pay for a chosen exhibition.

**You are required to**

1. Create UML Class diagrams for the **required instantiable classes**.
2. Implement the instantiable Java classes.
3. Use pseudocode to design an **application program**, using appropriate methods with suitable parameters.
4. Implement the application program.
5. Devise a suitable Test Plan to test the operation of the system.

Provide Harvard references for any books and/or websites that you may have used.

**Assessment Criteria**

|  |  |
| --- | --- |
| **K1** | **Recognise the relationship between objects and classes** |
| **K2** | **Understand and apply the key concepts of Object Oriented programming** |
| **I1** | **Use an appropriate technique to design, implement and document solutions to programming problems** |
| **I2** | **Demonstrate analysis in problem solving by breaking a complex problem down into smaller problems** |
| **P1** | **Assimilate and use basic object oriented concepts** |
| **T2** | **Manage own learning time effectively** |

The instantiable classes have been designed and their functions have been correctly programmed

* Appropriate UML diagrams have been provided **(15 marks)**

###### For each instantiable class, the student has

* declared all variables correctly **(6 marks)**
* coded relevant constructors **(9 marks)**
* coded suitable getters and setters **(6 marks)**
* programmed other methods correctly **(12 marks)**
* program layout consistent and code is tidy and commented **(6 marks)**

**Application Program**

* An appropriate pseudocode design has been provided for the application program

**(10 marks)**

* new instances of required instantiable classes have been correctly created

**(3 marks)**

* methods coded appropriately to carry out the required actions in a user-friendly way. Menu-based input used as necessary.

**(20 marks)**

* Layout and Commenting **(3 marks)**

|  |  |
| --- | --- |
| **P2** | **Create and apply appropriate test strategies** |

* A test plan has been designed and implemented and results have been documented appropriately **(8 marks)**

A Bibliography has been produced using Harvard Referencing. **(2 marks)**