|  |  |
| --- | --- |
| **Test Case No.** | **TD-001** |
| **Title** | **Create a blank class object** |
| **Description** | **Create a pure class object without attribute & method** |
| **Precondition** | **N/A** |
| **Assumption** | **N/A** |
| **Test Steps** | **Input the following data in the text area:**  **class Employee** |
| **Expected data structure** | **A UML class structure named “Employee” is created:**  ***class Employee {***  ***}*** |
| **Expected output drawing** |  |

|  |  |
| --- | --- |
| **Test Case No.** | **TD-002** |
| **Title** | **Create a class object with public attributes** |
| **Description** | **Create a class object with public attributes only** |
| **Precondition** | **N/A** |
| **Assumption** | **N/A** |
| **Test Steps** | **Input the following data in the text area:**  **class Employee**  **+ date\_of\_join : date**  **+ eng\_name : string**  **+ chi\_name : string** |
| **Expected data structure** | **A UML Class structure named “Employee” is created:**  ***class Employee {***  ***public date date\_of\_join;***  ***public string eng\_name;***  ***public string chi\_name;***  ***}*** |
| **Expected output drawing** |  |

|  |  |
| --- | --- |
| **Test Case No.** | **TD-003** |
| **Title** | **Create a class object with attributes of different modifiers** |
| **Description** | **Create a class object with private, protected, public and inner attributes** |
| **Precondition** | **N/A** |
| **Assumption** | **N/A** |
| **Test Steps** | **Input the following data in the text area:**  **class Employee**  **- date\_of\_join : date**  **+ eng\_name : string**  **+ chi\_name : string**  **# has\_dependent: bool**  **~ hkid\_no: string** |
| **Expected data structure** | **A UML Class structure named “Employee” is created:**  ***class Employee {***  ***private date date\_of\_join;***  ***public string eng\_name;***  ***public string chi\_name;***  ***protected bool has\_dependnet;***  ***string hkid\_no;***  ***}*** |
| **Expected output drawing** |  |

|  |  |
| --- | --- |
| **Test Case No.** | **TD-004** |
| **Title** | **Create a sub-class object** |
| **Description** | **Create a class object which extends another class** |
| **Precondition** | **N/A** |
| **Assumption** | **N/A** |
| **Test Steps** | **Input the following data in the text area:**  **class Employee**  **- date\_of\_join : date**  **- eng\_name : string**  **- chi\_name : string**  **- has\_dependent: bool**  **- hkid\_no: string**  **Class Worker >> Employee** |
| **Expected data structure** | **A UML Class structure named “Employee” is created:**  ***class Employee {***  ***private date date\_of\_join;***  ***private string eng\_name;***  ***private string chi\_name;***  ***private bool has\_dependnet;***  ***private string hkid\_no;***  ***}***  ***class Worker extends Employee {***  ***}*** |
| **Expected output drawing** |  |

|  |  |
| --- | --- |
| **Test Case No.** | **TD-005** |
| **Title** | **Create a sub-class object** |
| **Description** | **Create a class object which extends another class which does not exist** |
| **Precondition** | **N/A** |
| **Assumption** | **N/A** |
| **Test Steps** | **Input the following data in the text area:**  **Class Worker >> Employee** |
| **Expected data structure** | **A UML Class object named “Employee” is created:**  ***class Employee {***  ***}***  ***class Worker extends Employee {***  ***}*** |
| **Expected output drawing** |  |

|  |  |
| --- | --- |
| **Test Case No.** | **TD-006** |
| **Title** | **Create a class object with methods** |
| **Description** | **Create a class object with methods** |
| **Precondition** | **N/A** |
| **Assumption** | **N/A** |
| **Test Steps** | **Input the following data in the text area:**  **class Employee**  **- date\_of\_join : date**  **- eng\_name : string**  **- chi\_name : string**  **- salary : double**  **- hkid\_no: string**  **+ increaseSalary(double percentage): void**  **+ getEmployeeFullName(): string**  **- hasWorkingVisa (): bool** |
| **Expected data structure** | **A UML Class object named “Employee” is created:**  ***class Employee {***  ***private date date\_of\_join;***  ***private string eng\_name;***  ***private string chi\_name;***  ***private double salary;***  ***private string hkid\_no;***  ***public void increaseSalary(double percentage) {***  ***}***  ***public string getEmployeeFullName() {***  ***}***  ***private bool hasWorkingVisa() {***  ***}***  ***}*** |
| **Expected output drawing** |  |

|  |  |
| --- | --- |
| **Test Case No.** | **TD-007** |
| **Title** | **Create an interface object with methods** |
| **Description** | **Create an interface object with methods** |
| **Precondition** | **N/A** |
| **Assumption** | **N/A** |
| **Test Steps** | **Input the following data in the text area:**  **interface IExpatriate**  **+ applyWorkingVisa (date expirydate): boolean**  **+ claimExpense(date claim\_date, double claim\_amount): boolean**  **class Employee || IExpatriate**  **- date\_of\_join : date**  **- eng\_name : string**  **- chi\_name : string**  **- salary : double**  **- hkid\_no: string**  **+ increaseSalary(double percentage): void**  **+ getEmployeeFullName(): string** |
| **Expected data structure** | **A UML Class object named “Employee” with interface IExpatriate is created:**  ***interface IExpatriate {***  ***public boolean applyWorkingVisa(date expirydate);***  ***public boolean claimExpense(date claim\_date, double claim\_amount);***  ***}***  ***class Employee implements IExpatriate {***  ***private date date\_of\_join;***  ***private string eng\_name;***  ***private string chi\_name;***  ***private double salary;***  ***private string hkid\_no;***  ***public void increaseSalary(double percentage) {***  ***}***  ***public string getEmployeeFullName() {***  ***}***  ***public boolean applyWorkingVisa(date expirydate) {***  ***}***  ***public boolean claimExpense(date claim\_date, double claim\_amount) {***  ***}***  ***}*** |
| **Expected output drawing** |  |

|  |  |
| --- | --- |
| **Test Case No.** | **TD-008** |
| **Title** | **Create two interface objects with methods** |
| **Description** | **Create two interface objects with methods** |
| **Precondition** | **N/A** |
| **Assumption** | **N/A** |
| **Test Steps** | **Input the following data in the text area:**  **interface IExpatriate**  **+ applyWorkingVisa (date expirydate): boolean**  **+ claimExpense(date claim\_date, double claim\_amount): boolean**  **interface ITravelBenefit**  **+ assignDriver (string driver\_name): void**  **+ claimTravelExpense(date claim\_date, double claim\_amount): boolean**  **class Employee || IExpatriate || ITravelBenefit**  **- date\_of\_join : date**  **- eng\_name : string**  **- chi\_name : string**  **- salary : double**  **- hkid\_no: string**  **+ increaseSalary(double percentage): void**  **+ getEmployeeFullName(): string** |
| **Expected data structure** | **A UML Class object named “Employee” with interface IExpatriate and ITravelBenefit is created:**  ***interface IExpatriate {***  ***public boolean applyWorkingVisa(date expirydate);***  ***public boolean claimExpense(date claim\_date, double claim\_amount);***  ***}***  ***interface ITravelBenefit {***  ***public void assignDriver(string driver\_name);***  ***public boolean TravelExpense(date claim\_date, double claim\_amount);***  ***}***  ***class Employee implements IExpatriate, ITravelBenefit {***  ***private date date\_of\_join;***  ***private string eng\_name;***  ***private string chi\_name;***  ***private double salary;***  ***private string hkid\_no;***  ***public void increaseSalary(double percentage) {***  ***}***  ***public string getEmployeeFullName() {***  ***}***  ***public boolean applyWorkingVisa(date expirydate) {***  ***}***  ***public boolean claimExpense(date claim\_date, double claim\_amount) {***  ***}***  ***public void assignDriver(string driver\_name) {***  ***}***  ***public boolean TravelExpense(date claim\_date, double claim\_amount) {***  ***}***  ***}*** |
| **Expected output drawing** |  |