

**Abdul Basit**

**46658**

**BSCS\_2A-F22**

**OOP ASSIGNMENT (MIDS PAPER)**

**Question 1: Write the outputs of following code snippets.**

* **OUTPUT :**

ERROR

* **OUTPUT :**

Table Of Content is shown

Table Of Content is shown

* **OUTPUT :**

Error

* **OUTPUT :**

A’s constructor called

**Question 2: Give Short Questions of the following Questions :**

* **What is the purpose of access modifiers in OOP languages?**

**Public:** The public access modifier makes the class member accessible from anywhere in the program, including outside the class. Public members can be accessed and modified by any part of the program, which can make them useful for creating a public interface for a class.

**Private:** The private access modifier makes the class member accessible only within the same class where it is declared. Private members cannot be accessed from outside the class, including from any subclasses or other parts of the program. This helps to ensure that the internal state of the class is not modified in unexpected ways

**Protected:** The protected access modifier makes the class member accessible within the same class and any subclasses derived from it. Protected members cannot be accessed from outside the class hierarchy. This helps to ensure that the class can be extended safely without exposing its internal implementation details.

* **If we want to access the private members of a class in the child class what do we need to change?**

private members are not accessed outside the class.if we want to access the private members then we have to change its access specifers to protected or public.

* **Determine the accessibility of functions and data members in the following scenarios :**

|  |  |  |
| --- | --- | --- |
| **Scenarios** | **Accessible** | **Not Accessible** |
| **A private data member is declared in a class in accessible by its object in the main function.** |  | **Not Accessible** |
| **A Protected function defined in parent class by the functions of the child class.** | **Accessible** |  |
| **A public data member of the parent class by the object of child class** | **Accessible** |  |
|  |  |  |

**Ques 4 . Identify the errors.**

**CODE:**

**class B1{**

**public:**

**int i;**

**int j;**

**void g(int){**

**}**

**};**

**class B2{**

**public :**

**int j;**

**void g(){**

**}**

**};**

**class D: public B1, public B2 {**

**public:**

**int i;**

**};**

**int main()**

**{**

**D dobj;**

**D \*dptr=&dobj;**

**dptr->i=5;**

**dptr->i=10;**

**}**

**ERRORS:**

• i; in line 4. i was not declared

• class D: public B1; class public B2

• the inheritance should be on a seprate line by using ","

• dobj.g(); B2 has an empty body so calling a function in main

**Question 4 :**

**You have to develop a game that has multiple characters. These characters share some common properties like id, name, maximum power and strength. There are other properties as well that they have their own like Doremon has properties like a list of names gadgets and the name of partner, Benten has the watch Name, a list of names powers and total charge of the watch. There are also some common actions that they can perform like walk, jump and eat. Doremon can show Gadgets, launch attack and fly. Benten can perform the actions like rotate watch, fight and drive.**

**Implement the game using Inheritance in C++.**

**CODE :**

#include <iostream>

#include <string>

using namespace std;

class Character {

public:

int id;

string name;

int max\_power;

int strength;

Character(int id, string name, int max\_power, int strength) {

this->id = id;

this->name = name;

this->max\_power = max\_power;

this->strength = strength;

}

void walk() {

cout << name << " is walking." << endl;

}

void jump() {

cout << name << " is jumping." << endl;

}

void eat() {

cout << name << " is eating." << endl;

}

};

class Doremon : public Character {

public:

string gadgets[3];

string partner\_name;

Doremon(int id, string name, int max\_power, int strength, string gadgets[3], string partner\_name)

: Character(id, name, max\_power, strength) {

for (int i = 0; i < 3; i++) {

this->gadgets[i] = gadgets[i];

}

this->partner\_name = partner\_name;

}

void showGadgets() {

cout << "Gadgets: ";

for (int i = 0; i < 3; i++) {

cout << gadgets[i] << ", ";

}

cout << endl;

}

void launchAttack() {

cout << name << " is launching an attack!" << endl;

}

void fly() {

cout << name << " is flying." << endl;

}

};

class Benten : public Character {

public:

string watch\_name;

string powers[3];

int watch\_charge;

Benten(int id, string name, int max\_power, int strength, string watch\_name, string powers[3], int watch\_charge)

: Character(id, name, max\_power, strength) {

this->watch\_name = watch\_name;

for (int i = 0; i < 3; i++) {

this->powers[i] = powers[i];

}

this->watch\_charge = watch\_charge;

}

void rotateWatch() {

cout << "Rotating the " << watch\_name << " watch." << endl;

}

void fight() {

cout << name << " is fighting." << endl;

}

void drive() {

cout << name << " is driving." << endl;

}

};

int main() {

string gadgets[3] = {"Anywhere Door", "Takecopter", "Small Light"};

Doremon doremon(1, "Doremon", 100, 50, gadgets, "Nobita");

string powers[3] = {"Fireball", "Thunderbolt", "Big Bang"};

Benten benten(2, "Benten", 150, 75, "Omnitrix", powers, 50);

doremon.walk();

doremon.showGadgets();

doremon.launchAttack();

doremon.fly();

benten.jump();

benten.rotateWatch();

benten.fight();

benten.drive();

return 0;

}