

CAP – 202

Object Oriented Programming

(Theory)

Name: **Shubham Prakash**

Reg. No. : **11915803**

Submitted to: **Uminder Kaur**

Assignment – 01

(SET - 4)

1. Discuss the concept of private access specifier with the help of a program.

Solution: A private member variable or function cannot be accessed, or even viewed from outside the class. Only the class and friend functions can access private members. we define data in private section and related functions in public section so that they can be called from outside of the class. Private access specifier is used while creating a class, then the public and protected data members of the base class become the private member of the derived class and private member of base class remains private. When function which is not the member of class or friend function try to access a private member of that class it results in an error.

Example:

```
#include<iostream>
using namespace std;

class Circle
{
    // private data member
    private:
        double radius;

    // public member function
    public:
        double compute_area()
        { // member function can access private
          // data member radius
        }
```

```

        return 3.14*radius*radius;
    }
};

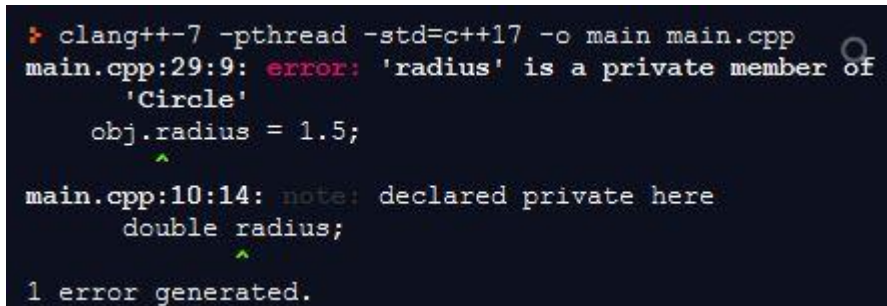
// main function
int main()
{
    // creating object of the class
    Circle obj;

    // trying to access private data member
    // directly outside the class
    obj.radius = 1.5;

    cout << "Area is:" << obj.compute_area();
    return 0;
}

```

Output :



```

❏ clang++-7 -pthread -std=c++17 -o main main.cpp
main.cpp:29:9: error: 'radius' is a private member of
      'Circle'
    obj.radius = 1.5;
        ^
main.cpp:10:14: note: declared private here
    double radius;
        ^
1 error generated.

```

Output gives error, as we cannot access the data when it is in private access specifier. So when we change the private member to public we get the right output.

```

#include<iostream>
using namespace std;

class Circle
{
    // public data member
    public:
        double radius;

    public:
        double compute_area()
        {
            return 3.14*radius*radius;
        }
}

```

```

    }
};

int main()
{
    Circle obj;
    obj.radius = 1.5;
    cout << "Area is:" << obj.compute_area();
    return 0;
}

```

Output:

```

❖ clang++-7 -pthread -std=c++17 -o main main.cpp
❖ ./main
Area is:7.065❖ □

```

2. Write a short note on Keywords and identifiers.

Solution:

KEYWORD:

- A keyword is a reserved word.
- They are already defined in the header library.
- You cannot use it as a variable name, constant name or any other.
- A list of 32 Keywords in C++ Language which are also available in C language.

Example:

- class
- public
- private
- cout.

IDENTIFIERS:

- The C++ identifier is a name used to identify a variable, function, class, module, or any other user-defined item.
- An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores, and digits (0 to 9).
- C++ does not allow punctuation characters such as @, \$, and % within identifiers.
- C++ is a case-sensitive programming language.

Example:

- Fname
- mydob
- anythingb
- ram.

3. Discuss the concept of function overloading with the help of an example.

Solution:

Function overloading is a C++ programming feature that allows us to have more than one function having same name but different parameter list, when I say parameter list, it means the data type and sequence of the parameters, for example the parameters list of a function myfuncn(int a, float b) is (int, float) which is different from the function myfuncn(float a, int b) parameter list (float, int).

Example:

```
#include <iostream>
using namespace std;
class laalu{
public:
void print(int i) {
    cout << " Here is int " << i << endl;
}
void print(double f) {
    cout << " Here is float " << f << endl;
}
void print(char const *c) {
    cout << " Here is char* " << c << endl;
}
};
int main() {
    laalu la;
    la.print(10);
    la.print(10.10);
    la.print("ten");
    return 0;
}
```

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
Here is int 10
Here is float 10.1
Here is char* ten
> □
```

4. Differentiate between structure and class.

Solution:

STRUCTURES in C++

- A value datatype that can hold related data that belongs to various data types.
- Cannot inherit other classes.
- An instance is a structure variable.
- The keyword to define is “struct”.
- If there are no access specifier declared, then the member or the properties are public.

CLASS in C++

- A standard format that defines data and methods to create objects.
- Can inherit other classes or structures.
- An instance is an object
- The keyword to define is “class”.
- If there are no access specifiers declared, then the members are private by default.

Thanking You

Shubham Prakash

11915803