

ANS 1:

(a) The main difference between a struct and a class in C++ is that for a structure all members are public by default, whereas for a class they are private by default. Structures are generally used for grouping simple data while class is used for implementing OOP concept like abstraction. Both structures and class can have constructors, destructors and member functions, they can be used with inheritance.

(b)

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

```
class State_Bank {
```

```
private:
```

```
String name;
int acc_no;
float bal;
```

```
public:
```

```
    static float total_balance;
```

```
    State_Bank (String n, int a, float b) {
```

```
        name = n;
        acc_no = a;
        bal = b;
        total_balance += b;
```

```
}
```

```
void get_details () {
```

```
    cout << "Name:" << name << endl
        << "Account number:" << acc_no << endl
        << "Balance:" << bal << endl;
```

```
cout << "Total balance is :" << total - balance << endl;
```

4

};

```
float State_Bank::total - balance = 0;
```

```
struct State_Bank_Branch {
```

```
String name;
```

```
int account - no;
```

```
float - bal;
```

```
Void get - details () {
```

```
cout << "Name :" << name << endl;
```

```
<< "Account number :" << acc - no << endl;
```

```
<< "Balance :" << bal << endl;
```

4

};

```
int main () {
```

```
State_Bank c1 ("Shruthika", 23567, 4000.0);
```

```
c1.get - details ();
```

```
State_Bank_Branch b1 = {"Sadhika", 1111, 5000.0};
```

```
b1.get - details ();
```

```
return 0;
```

4

ANS 2:

(a) Public members can be accessed from anywhere outside the class, including from other functions and classes. Private members can only be accessed from within the same class, it helps in data hiding and encapsulation.

(b)

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

```
class Student {
```

public :

```
    string name;
```

private :

```
    int age;
```

```
};
```

```
int main() {
```

```
    Student s;
```

```
    s.name = "Shruthika";
```

```
    cout << "Name : " << s.name << endl;
```

```
    return 0;
```

```
}
```

ANS 8 :

(a) In OOP, a method is a type of function that is associated with a specific class or object. It defines the behaviour of that object and has direct access to the object's internal data. In general programming, a function is a block of code that performs a specific task, it is a stand-alone unit of code and can be called anywhere in the program. It may take inputs and return outputs.

(b)

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

void hello hello_function() {
    cout << "Hello" << endl;
}

class Example {
public:
    void hello_method() {
        cout << "Hello method" << endl;
    }
};

int main() {
    hello_function();
    Example e;
    e.hello_method();
    return 0;
}
```

ANS 4:

- (a) A constructor is a special member function of a class that is automatically called when an object of that class is created. It initializes the object.
- (b) A default constructor is a constructor that takes no arguments whereas a parameterized constructor is a constructor that takes one or more arguments.

```

#include <iostream>
using namespace std;

class Dot {
    int x, y;

public:
    Dot() {x=0; y=0; }
    Dot (int a) {x=a; y=0; }
    Dot (int a, int b) {x=a; y=b; }

    void get_details() {
        cout << "(" << x << ";" << y << ")";
    }
};

int main () {
    Dot d1, d2(5), d3(3,4);
    d1.get_details();
    d2.get_details();
    d3.get_details();
    return 0;
}

```

Ans 5:

(a) A static variable is a special type of variable that has a single copy, regardless of how many objects of the class are created. It is shared by all objects of that class.

```

(b) #include <iostream>
using namespace std;

class Counter {
    static int count;

public:
    Counter() {
        count++;
    }
}

```

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

```
class car {
```

```
public:
```

```
    string brand;
    int year;
```

```
car() { // DEFAULT CONSTRUCTOR
```

```
    brand = "Don't know";
```

```
    year = 0;
```

```
}
```

```
car(string b, int y) { // PARAMETERIZED CONSTRUCTOR
```

```
    brand = b;
```

```
    year = y;
```

```
}
```

```
void get_details()
```

```
    cout << brand << "(" << year << ")" << endl;
```

```
};
```

```
int main()
```

```
    car c1;
```

```
    car c2("Honda", 2022);
```

```
    c1.get_details();
```

```
    c2.get_details();
```

```
    return 0;
```

(c) Constructor overloading is the process of having multiple constructors in a class with the same name but different parameter lists.

```

Void Show() {
    cout << "Count:" << count << endl;
}

int Counter::Count = 0;

int main() {
    Counter c1, c2 (3);
    c1.Show();
    c2.Show();
    c3.Show();
    return 0;
}

```

Ans 6:

(a) A friend function is a function that is not a member of a class but has special permission to access the private and protected members of that class. A friend function is needed when a function that is external to a class needs to access its private data members.

```

(b) #include <iostream>
using namespace std;

class Box {
    int length;

public:
    Box(int l) {length = l;} }

    friend void get_Length(Box b);

}

Void get_Length(Box b) {
    cout << "Length:" << b.length << endl;
}

int main() {
    Box b(10);
    get_Length(b);
    return 0;
}

```