C++

1-why reference variables are safer than pointer variables and which is powerful ?

Ans – the reason why reference variables are safer than pointers are the reference variables needs to be initialised with some normal variable while declaring. So there is no chances of illegal memory access in reference variables. The pointer varialbes are powerful because it has the power to hold the address of different variables in its life time . but in in reference you can assign the new value to it it is constant.

2-why we have to store the address of particular data type to a same data-type pointer? How much size does a pointer variables takes?

Ans- the reason is to access the correct memory block . when we increment in pointer varialbes it gets increment in the address inside the pointer variable according to its size like if we increment +1 in pointer variable of int type then

Its value will become address+sizeof(int) and it will start pointing to that particular location .

The pointer variables takes 8bytes in 64bit os.

3- Why there is need of object oriented programming and how object oriented concept is implemented in c++?

Ans- The need of object oriented is very important especially in big projects where we have to write lot of code. Oops provide features like encapsulation , data hiding etc. this makes the programming very secure, error-free, and easy readability. The oops concept can be implemented using class in c++.

4- why recursion sometime can be very memory consumable?

Ans- The recursion is a process of a function calling itself. As we know functions gets the memory when its called so in recursion the function calls itself serval times until it hits the base condition. That is why its very memory consumable when we were doing big tasks with recursion.

5- How many constructures does a compiler makes itself if we don’t define explicitly ?

Ans – compiler makes two construtures 1. Default constructure 2 copy construture.

6-how compile-time polymorphism is solved by the compilers?

Ans- finding exact match/meaning by seeing its context at the time of compilation is called as compile time polymorphism, compiler solve this problem using early binding method in this method the compiler maps the function calling with their function definitions.

7- what is operator overloading and why do we need to do it?

Ans- operator overloading is a method to defining the meaning for the operator if it doesn’t know how to do it. We need to do it when we have to perform operator related tasks of user-defined data types.

8-is there’s a difference between \*ptr and ptr[0]?why?

Ans- there’s no difference in these. The inner working of [] is \*(ptr+0) which is same as \*ptr. We can write any of them according to your understanding.

9-how data-hiding concept protects from programme crashed?

Ans- data-hiding is very useful concept of oops it is used to protect data members to directly accessible by any function . only member functions and friend functions have the rights to access that data. It protects the data to be directly accessible this protect the data members to have corrupted data mistakenly entered by a programmer. So if there is no corrupted data there is no chance of programme to be crashed.

10-what is inline? Is it a request or a order for a compiler? how compiler decides which function are eligible for inline?

Ans- inline is a keyword which tells the compiler to puts the function code replace with function calling to save calling time. It is a request for the compiler. the compiler checks the functions requested by the programmer to make it inline by seeing its line of code and checking if there is loops in the function or not , if it finds a loop in the function it decline the request to make it inline otherwise it will make it inline if it meets the criterias to make it inline.