Group Discussion

Object Oriented Programming

Name: Hari Krishna Shah

VIT ID: 21BCS0167

Ques. Discuss the pros and cons of: Usage of reference variable in C++.

Introduction

A reference variable is an alias, that is, another name for an already existing variable. Once a reference is initialized with a variable, either the variable name or the reference name may be used to refer to the variable.

Syntax:

int i = 20;

int &r = i; // Here, r is a reference variable of integer i.

Usage of reference variable

1. Modify the passed parameters in a function: If a function receives a reference to a variable, it can modify the value of the variable. For example, the following program variables are swapped using references.

   Sample code

```
#include <iostream>
using namespace std;
void swap(int& first, int& second)

{
      int temp = first;
      first = second;
      second = temp;
}

int main()
{
      int a = 2, b = 3;
      swap(a, b);
      cout << a << " " << b;
      return 0;
}

Output: 3 2
```

2. Reference variable are used in copy constructor to make copy of objects.

   Sample code

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

class Point {
private:
    int x, y;

public:
    Point(int x1, int y1)
    {
        x = x1;
        y = y1;
    }

    // Copy constructor
    Point(const Point& p1)
    {
        x = p1.x;
        y = p1.y;
    }
};
```

3. Avoiding a copy of large structures: Imagine a function that has to receive a large object. If we pass it without reference, a new copy of it is created which causes wastage of CPU time and memory. We can use references to avoid this.

   Sample code:

```
struct Student {
       string name;
       string address;
       int rollNo;
}

// If we remove & in below function, a new
// copy of the student object is created.
// We use const to avoid accidental updates
// in the function as the purpose of the function
// is to print s only.
void print(const Student &s)
{
       cout << s.name << " " << s.address << " " << s.rollNo
             << '\n';
}
```

4. In For Each Loop to modify all objects: We can use references in for each loop to modify all elements.

   Sample code

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
        vector<int> vect{ 10, 20, 30, 40 };

        // We can modify elements if we
        // use reference
        for (int& x : vect) {
                x = x + 5;
        }

        // Printing elements
        for (int x : vect) {
                cout << x << " ";
        }
        cout << '\n';

        return 0;
}
```

Pros of reference variable

1. A reference variable does not consume any extra memory. It has the same memory address as the variable it refers to. While a pointer needs extra space for itself.
2. In order to access the value of the referenced variable using pointer, we need to use a dereferencing operator(*) whereas we can do the same using reference variable without any dereferencing method.
3. Reference variables are cleaner and modish as compare to the pointers; they can also be used while passing in the function as arguments, known as call by references.
4. References are safer and easier to use because they are referenced to a particular variable unlike pointer which can be made to point to different variables at different time.
5. No copy of arguments make pass by reference, faster to process.

6. The change in reference variable is reflected in the referenced variable.

Cons of reference variable

1. Readability. A person reading the code has no way of knowing that the value can be modified in the function.

For example: if int &x = i;

If the user changes doesn't know that variable x is a reference variable and the user changes the value of reference variable x, then the user may not know that the value of variable i will also change.

2. You can't reassign the reference variable to reference any other variable after the first reference.

For example:

Int &x = a;

Now, reference variable x will reference to the variable a in the life time of program.

3. Reference variable can't be NULL.

4. The pointer variable has n-levels/multiple levels of indirection i.e. single-pointer, double-pointer, triple-pointer. Whereas, the reference variable has only one/single level of indirection.