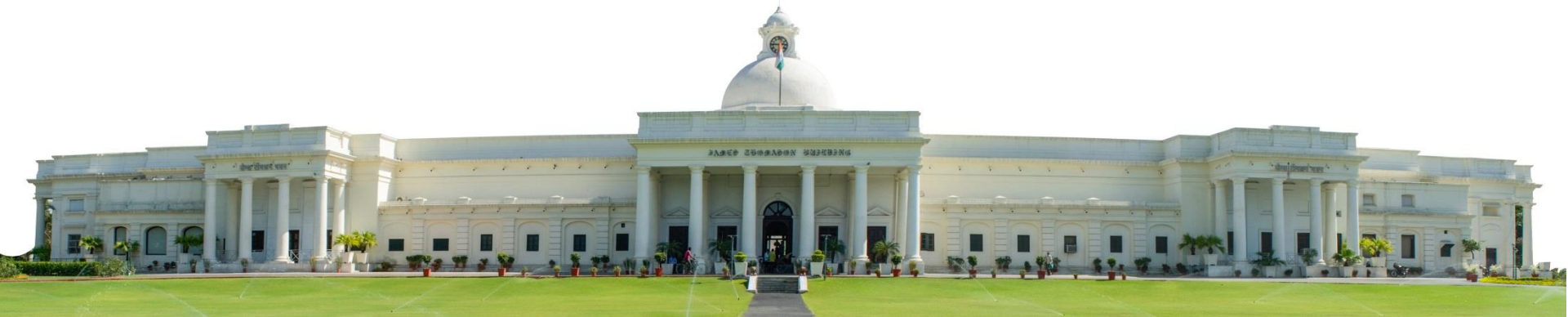
Programming with C and C++

CSC-101 (Lecture 32)

Dr. R. Balasubramanian
Professor
Department of Computer Science and Engineering
Mehta Family School of Data Science and Artificial Intelligence
Indian Institute of Technology Roorkee
Roorkee 247 667

bala@cs.iitr.ac.in
https://faculty.iitr.ac.in/cs/bala/



Example 2

- write a C++ program using class and objects to add two distances given in meter and centimeter

```
1   #include <iostream>
2
3   using namespace std;
4
5   class Distance {
6   private:
7       int meters;
8       int centimeters;
9
10  public:
11      void getDistance() {
12          cout << "Enter meters: ";
13          cin >> meters;
14          cout << "Enter centimeters: ";
15          cin >> centimeters;
16      }
17
```

https://ideone.com/ECT4fj

```
18      void displayDistance() {
19          cout << "Distance: " << meters << " meters " << centimeters
20          << " centimeters" << endl;
21      }
22
23      Distance addDistances(const Distance& d1, const Distance& d2) {
24          Distance result;
25          result.meters = d1.meters + d2.meters;
26          result.centimeters = d1.centimeters + d2.centimeters;
27
28          if (result.centimeters >= 100) {
29              result.meters += result.centimeters / 100;
30              result.centimeters = result.centimeters % 100;
31          }
32
33          return result;
34      }
35  };
```

```
36
37  int main() {
38      Distance distance1, distance2, result;
39
40      cout << "Enter the first distance:" << endl;
41      distance1.getDistance();
42      cout << "Enter the second distance:" << endl;
43      distance2.getDistance();
44
45      result = result.addDistances(distance1, distance2);
46
47      cout << "Sum of the distances:" << endl;
48      result.displayDistance();
49
50      return 0;
51  }
52
```

stdin

```
10 95
20 85
```

stdout

```
Enter the first distance:
Enter meters: Enter centimeters: Enter the second distance:
Enter meters: Enter centimeters: Sum of the distances:
Distance: 31 meters 80 centimeters
```

C++ Strings

```
1  #include <iostream>
2  using namespace std;
3  int main( ) {
4      string s1 = "CSC-101, IITR";
5          char ch[] = { 'C', '+', '+'};
6          string s2 = string(ch);
7          cout<<s1<<endl;
8          cout<<s2<<endl;
9  }
10
```

https://ideone.com/dOwwIj

stdout

```
CSC-101, IITR
C++
```

C++ Strings

https://ideone.com/UEt4HY

```
1   #include <iostream>
2   #include <string>
3
4   using namespace std;
5
6   int main () {
7
8       string str1 = "CSC-101, ";
9       string str2 = "IIT Roorkee";
10      string str3;
11      int  len ;
12
```

C++ Strings

```
13      // copy str1 into str3
14      str3 = str1;
15      cout << "str3 : " << str3 << endl;
16
17      // concatenates str1 and str2
18      str3 = str1 + str2;
19      cout << "str1 + str2 : " << str3 << endl;
20
21      // total length of str3 after concatenation
22      len = str3.size();
23      cout << "str3.size() :  " << len << endl;
24
25      return 0;
26  }
```

stdout

```
str3 : CSC-101,
str1 + str2 : CSC-101, IIT Roorkee
str3.size() :  20
```

Range-based for loop in C++

```
1  #include <iostream>
2  using namespace std;
3  int main ()
4  {
5  int IntArr [5] = { 100, 200, 300, 400, 500};
6  double DoubleArr [5] = { 1.5, 2.5, 3.5, 4.5, 5.5 };
7   // use range based for loop
8  for ( auto &var : IntArr )  //auto keyword
9  {
10 cout << var << " ";
11 }
12 cout <<endl;
13 for ( auto &var : DoubleArr )
14 {
15 cout << var << " " ;
16 }
17 return 0;
18 }
19
```

https://ideone.com/bwnqcP

stdout

```
100 200 300 400 500
1.5 2.5 3.5 4.5 5.5
```

auto in C++

https://ideone.com/Ux2ZTi

```
#include <iostream>
using namespace std;
int main() {

    auto a=45;
    auto b='z';
    auto c=10.3;

    cout<<"data type of the variable a is: "<<a<<endl;
    cout<<"data type of the variable b is: "<<b<<endl;
    cout<<"data type of the variable c is: "<<c<<endl<<endl;

    cout<<"data type of the variable a is: "<<sizeof(a)<<endl;
    cout<<"data type of the variable b is: "<<sizeof(b)<<endl;
    cout<<"data type of the variable c is: "<<sizeof(c)<<endl;
    return 0;
}
```

stdout

```
data type of the variable a is: 45
data type of the variable b is: z
data type of the variable c is: 10.3

data type of the variable a is: 4
data type of the variable b is: 1
data type of the variable c is: 8
```

C++ Constructor

- Constructor is a special method which is invoked automatically at the time of object creation.
- It is used to initialize the data members of new object generally.
- The constructor in C++ has the same name as class or structure.

```
1. <class-name> (list-of-parameters);

2. <class-name> (list-of-parameters)
  { // constructor definition }
```

- The following syntax is used to define a constructor outside of a class:

```
<class-name>: :<class-name> (list-of-parameters){ // constructor definition}
```

- **T**wo types of constructors in C++
  - Default constructor
  - Parameterized constructor

Default Constructor

https://ideone.com/3vupMw

```
1  #include <iostream>
2  using namespace std;
3  class Employee
4   {
5     public:
6          Employee()
7          {
8              cout<<"Testing Default Constructor"<<endl;
9          }
10 };
11 int main(void)
12 {
13     Employee e1; //creating an object of Employee
14     Employee e2;
15     return 0;
16 }
17
```

stdout

```
Testing Default Constructor
Testing Default Constructor
```

Default Constructor

- Design a class called Book with attributes for the title (string), author (string), and price (double). Implement a default constructor that sets the title and author to "Unknown" and the price to 0. Write a program to create an instance of the Book class using the default constructor and display its attributes.

```
1   #include <iostream>
2   #include <string>
3
4   using namespace std;
5
6   class Book {
7   private:
8       string title;
9       string author;
10      double price;
11
```

https://ideone.com/U0tLKk

```
12  public:
13      // Default constructor
14      Book() : title("Unknown"), author("Unknown"), price(0.0) {}
15
16      // Member function to display book details
17      void displayBookDetails() {
18          cout << "Title: " << title << endl;
19          cout << "Author: " << author << endl;
20          cout << "Price: Rs." << price << endl;
21      }
22  };
23
```

```
23
24  int main() {
25      // Create an instance of the Book class using the default constructor
26      Book myBook;
27
28      // Display the book details
29      cout << "Book Details:" << endl;
30      myBook.displayBookDetails();
31
32      return 0;
33  }
34
```

stdout

```
Book Details:
Title: Unknown
Author: Unknown
Price: Rs.0
```

Parameterized Constructor

https://ideone.com/p6kzuK

```
1  #include <iostream>
2  using namespace std;
3  class Employee {
4     public:
5         int id;//data member
6         string name;//data member
7         float salary;
8         Employee(int i, string n, float s)
9          {
10             id = i;
11             name = n;
12             salary = s;
13         }
14        void display()
15         {
16             cout<<id<<"  "<<name<<"  "<<salary<<endl;
17         }
18 };
```

Parameterized Constructor

```
19  int main(void) {
20      Employee e1 =Employee(18, "Virat", 100000);
21      Employee e2=Employee(45, "Rohit", 90000);
22      e1.display();
23      e2.display();
24      return 0;
25  }
```

stdout

```
18  Virat  100000
45  Rohit  90000
```

Parameterized Constructor

- Design a class Circle with a parameterized constructor that takes the radius as an argument and calculates the area of the circle in C++.

Thank You!