**Assignment 10**

1. **Default Constructor**

**Code:**

**#include <iostream>**

**using namespace std;**

**class Base {**

**public:**

**};**

**class A {**

**public:**

**A() { cout << "A Constructor" << endl; }**

**int n;**

**};**

**class B : public A {**

**};**

**class C : public A {**

**public:**

**C()**

**{**

**cout << "C Constructor" << endl;**

**}**

**};**

**class D {**

**public:**

**D()**

**{**

**cout << "D Constructor" << endl;**

**}**

**private:**

**A a;**

**};**

**int main()**

**{**

**Base ba;**

**B b;**

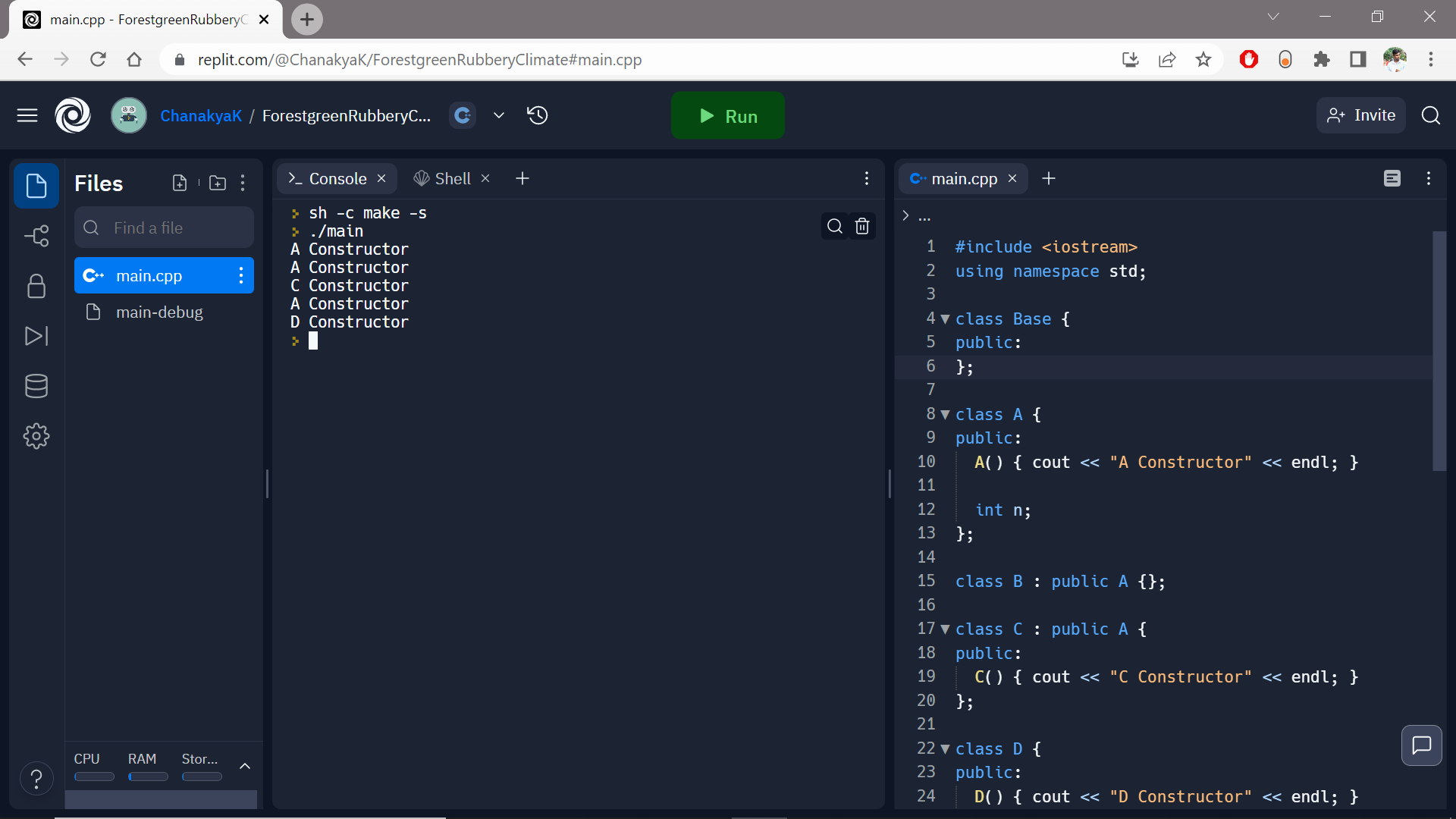
**C c;**

**D d;**

**return 0;**

**}**

**Output:**



1. **Parameterized Constructor**

**#include <iostream>**

**using namespace std;**

**class employee {**

**int empID;**

**char name[50];**

**double salary;**

**public:**

**employee();**

**void display();**

**};**

**employee::employee()**

**{**

**cout << "Enter the Employee ID:";**

**cin >> empID;**

**cout << "Enter the Name:";**

**cin >> name;**

**cout << "Enter the Salary:";**

**cin >> salary;**

**}**

**void employee::display()**

**{**

**cout << endl << empID << "\t" << name << "\t" << salary;**

**}**

**int main()**

**{**

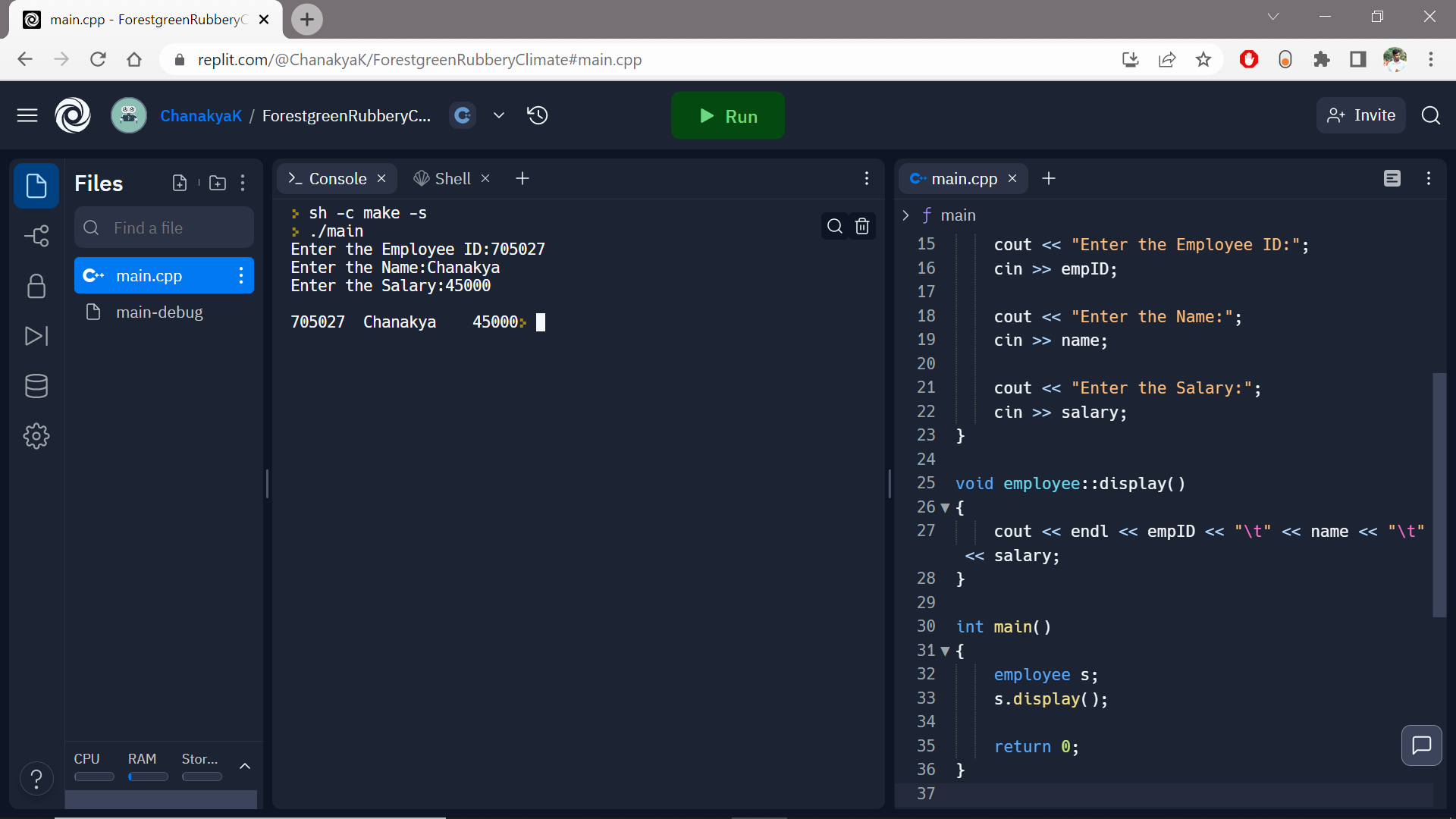
**employee s;**

**s.display();**

**return 0;**

**}**

**Output:**



1. **Copy Constructor**

**Code:**

**#include <iostream>**

**using namespace std;**

**class Copy {**

**private:**

**int x, y;**

**public:**

**Copy(int x1, int y1)**

**{**

**x = x1;**

**y = y1;**

**}**

**Copy(const Copy& c1)**

**{**

**x = c1.x;**

**y = c1.y;**

**}**

**int getX() { return x; }**

**int getY() { return y; }**

**};**

**int main()**

**{**

**Copy c1(10, 15);**

**Copy c2 = c1;**

**cout << "c1.x = " << c1.getX()**

**<< ", c1.y = " << c1.getY();**

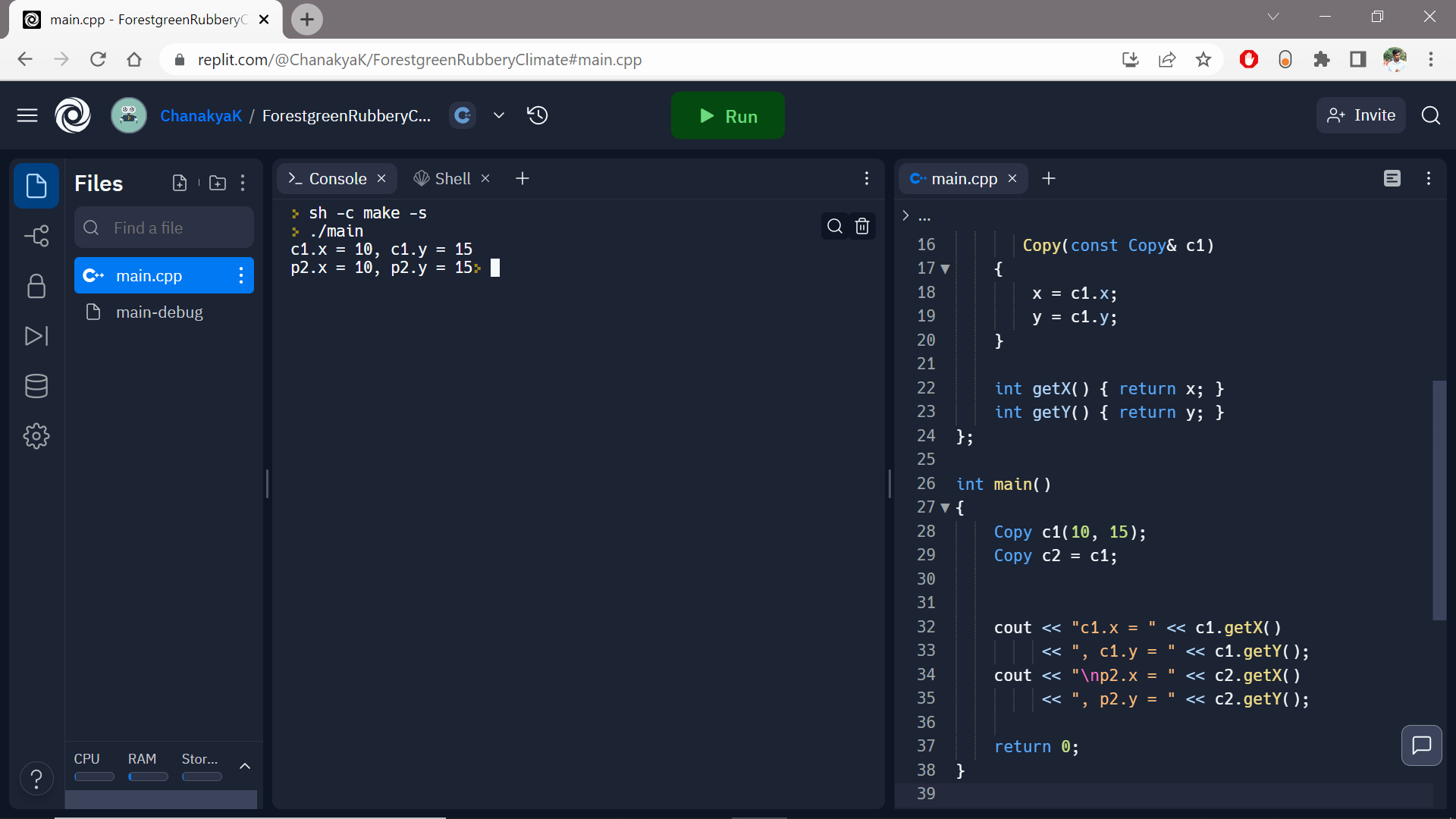
**cout << "\np2.x = " << c2.getX()**

**<< ", p2.y = " << c2.getY();**

**return 0;**

**}**

**Output:**



1. **Function Overloading**

**Code:**

**#include<iostream>**

**using namespace std;**

**void mul(int a, double b)**

**{**

**cout<<"product = "<<(a\*b);**

**}**

**void mul(double a, int b)**

**{**

**cout<<endl<<"product = "<<(a\*b);**

**}**

**int main()**

**{**

**mul(3,9.5);**

**mul(6.5,60);**

**return 0;**

**}**

**Output:**

Graphical user interface, text

Description automatically generated

1. **Destructor**

**Code:**

**#include<iostream>**

**using namespace std;**

**int count=0;**

**class Dtest**

**{**

**public:**

**Dtest()**

**{**

**count++;**

**cout<<"\n No. of Object created:\t"<<count;**

**}**

**~Dtest()**

**{**

**cout<<"\n No. of Object destroyed:\t"<<count;**

**--count;**

**}**

**};**

**int main()**

**{**

**Dtest d,d1,d2,d3;**

**return 0;**

**}**

**Output:**

Text

Description automatically generated with low confidence