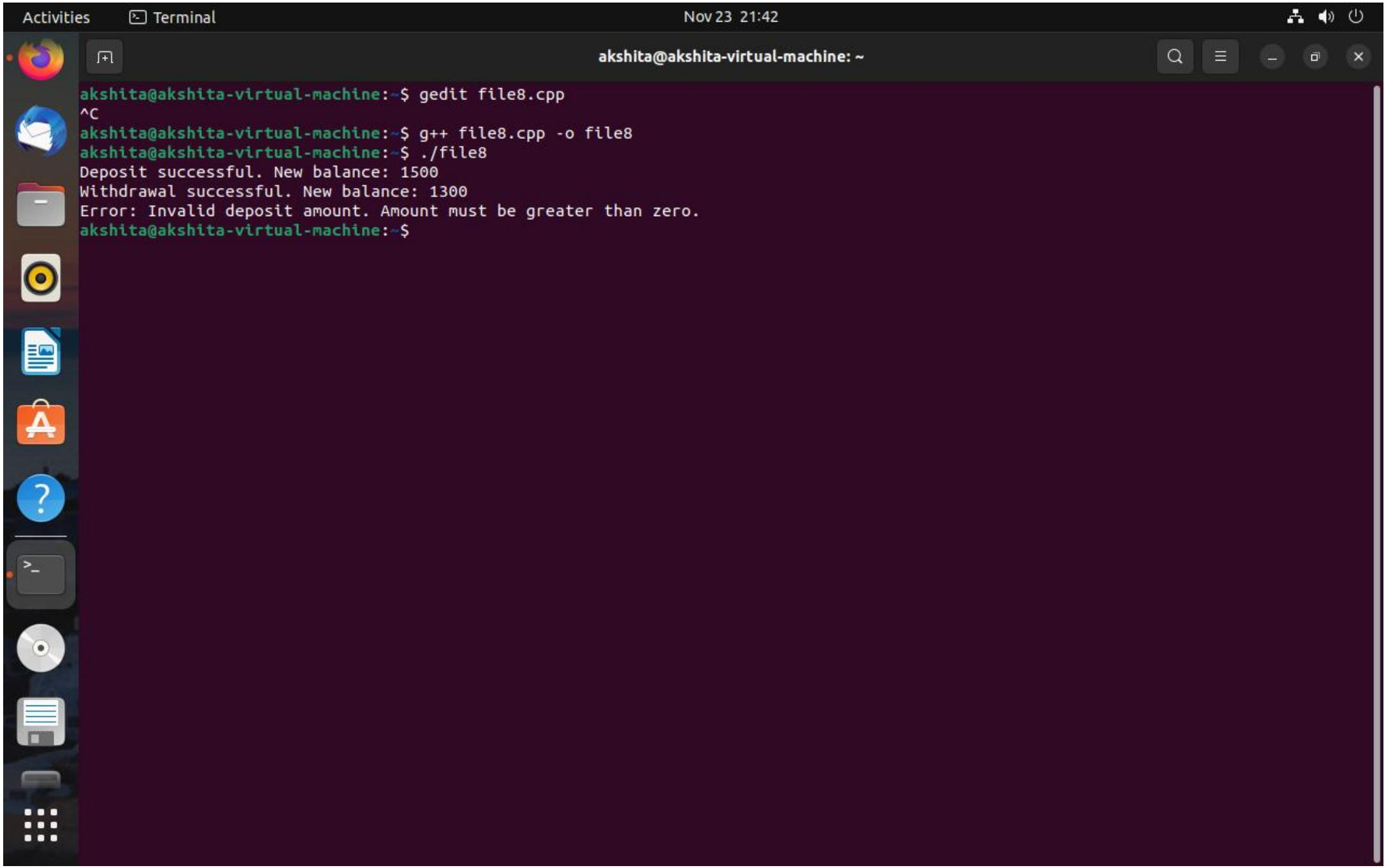
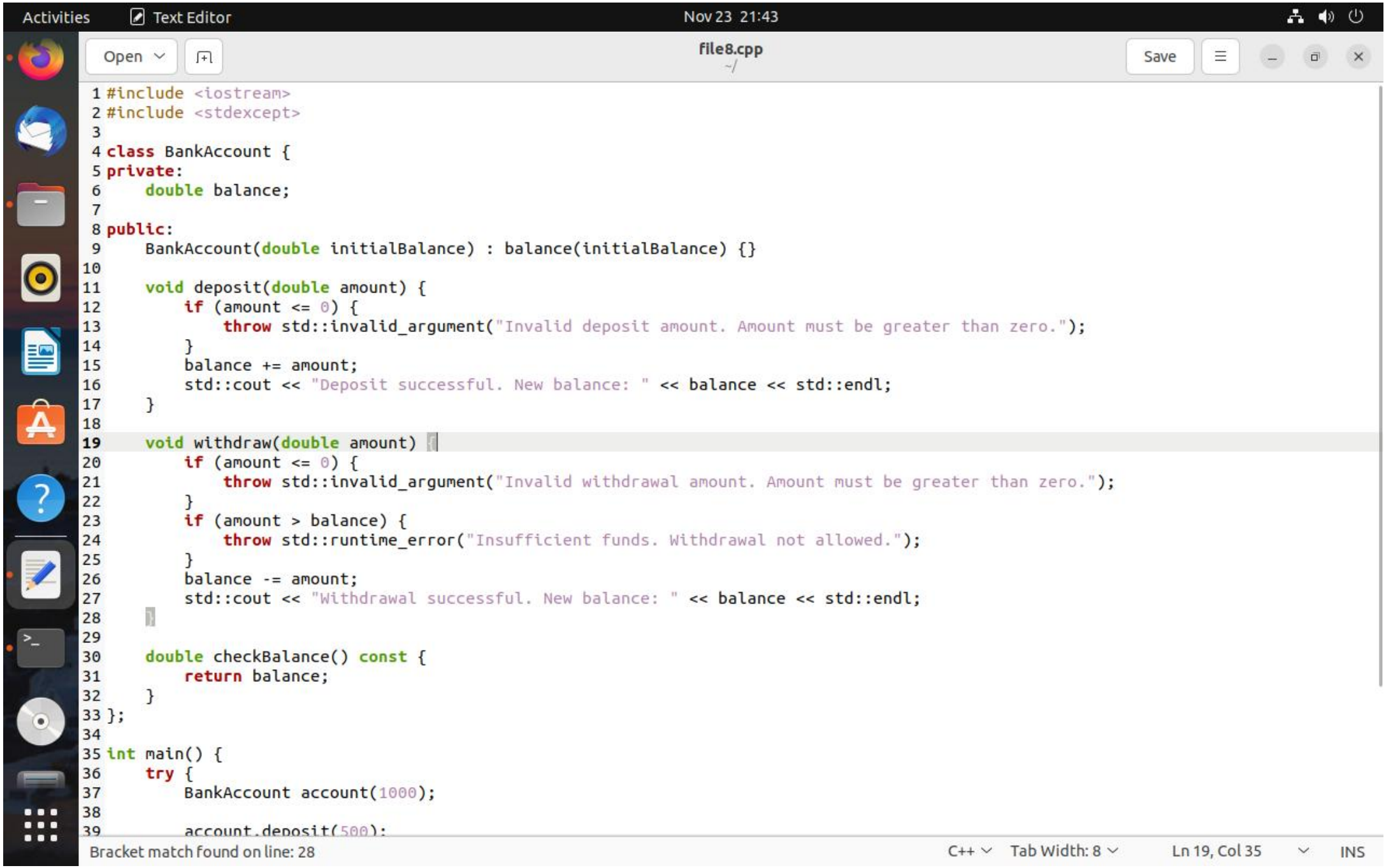
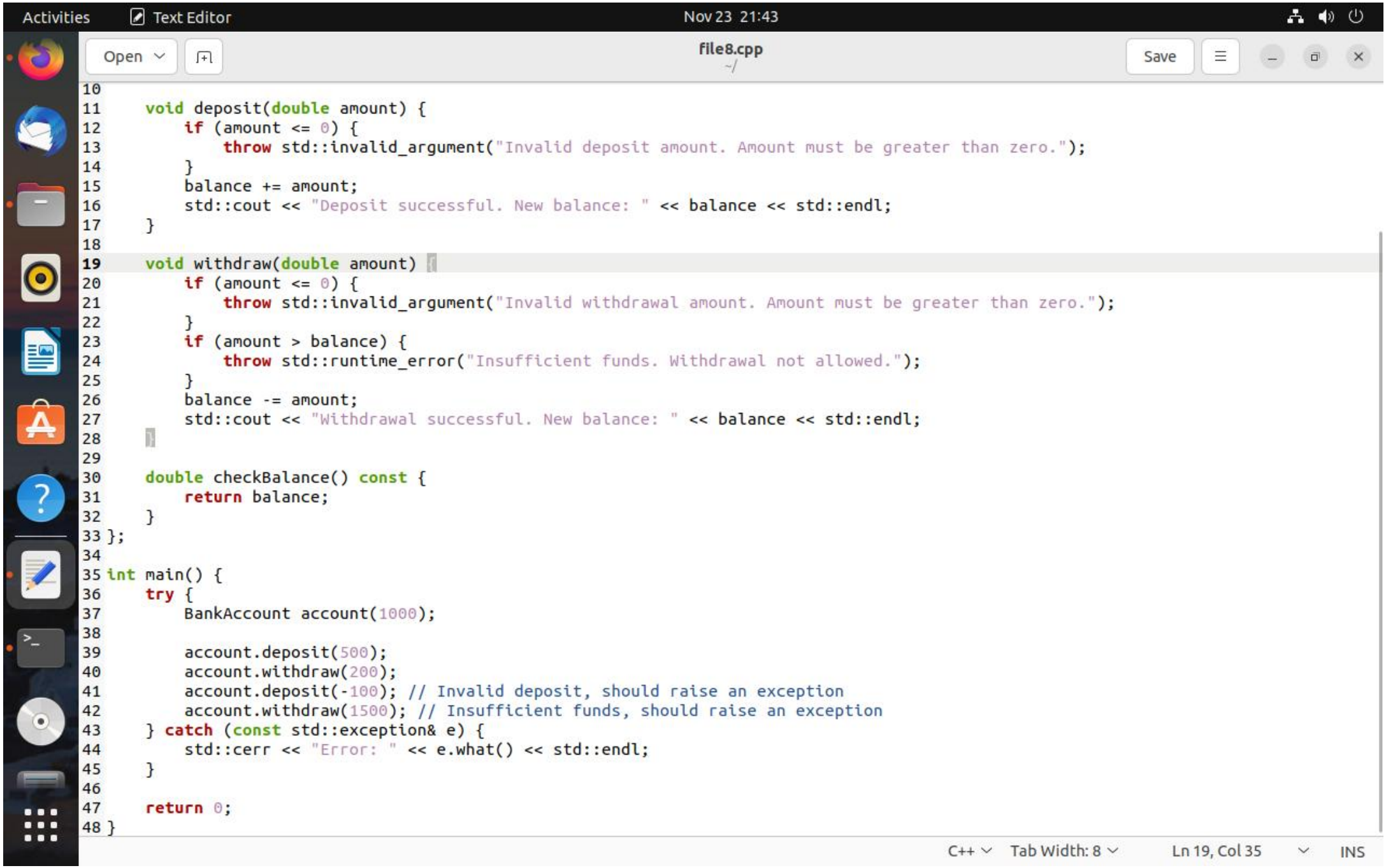

Activities
Terminal
Nov 23 21:42
akshita@akshita-virtual-machine: ~
akshita@akshita-virtual-machine:~$ gedit file8.cpp
^C
akshita@akshita-virtual-machine:~$ g++ file8.cpp -o file8
akshita@akshita-virtual-machine:~$ ./file8
Deposit successful. New balance: 1500
Withdrawal successful. New balance: 1300
Error: Invalid deposit amount. Amount must be greater than zero.
akshita@akshita-virtual-machine:~$


Activities
Text Editor
Nov 23 21:43
Open
file8.cpp
~/
Save
#include <iostream>
#include <stdexcept>
class BankAccount {
private:
double balance;
public:
BankAccount(double initialBalance) : balance(initialBalance) {}
void deposit(double amount) {
if (amount <= 0) {
throw std::invalid_argument("Invalid deposit amount. Amount must be greater than zero.");
}
balance += amount;
std::cout << "Deposit successful. New balance: " << balance << std::endl;
}
void withdraw(double amount) {
if (amount <= 0) {
throw std::invalid_argument("Invalid withdrawal amount. Amount must be greater than zero.");
}
if (amount > balance) {
throw std::runtime_error("Insufficient funds. Withdrawal not allowed.");
}
balance -= amount;
std::cout << "Withdrawal successful. New balance: " << balance << std::endl;
}
double checkBalance() const {
return balance;
}
};
int main() {
try {
BankAccount account(1000);
account.deposit(500);
Bracket match found on line: 28
C++
Tab Width: 8
Ln 19, Col 35
INS


Activities
Text Editor
Nov 23 21:43
Open
file8.cpp
~/
Save
10
11 void deposit(double amount) {
12 if (amount <= 0) {
13 throw std::invalid_argument("Invalid deposit amount. Amount must be greater than zero.");
14 }
15 balance += amount;
16 std::cout << "Deposit successful. New balance: " << balance << std::endl;
17 }
18
19 void withdraw(double amount) {
20 if (amount <= 0) {
21 throw std::invalid_argument("Invalid withdrawal amount. Amount must be greater than zero.");
22 }
23 if (amount > balance) {
24 throw std::runtime_error("Insufficient funds. Withdrawal not allowed.");
25 }
26 balance -= amount;
27 std::cout << "Withdrawal successful. New balance: " << balance << std::endl;
28 }
29
30 double checkBalance() const {
31 return balance;
32 }
33 };
34
35 int main() {
36 try {
37 BankAccount account(1000);
38
39 account.deposit(500);
40 account.withdraw(200);
41 account.deposit(-100); // Invalid deposit, should raise an exception
42 account.withdraw(1500); // Insufficient funds, should raise an exception
43 } catch (const std::exception& e) {
44 std::cerr << "Error: " << e.what() << std::endl;
45 }
46
47 return 0;
48 }
C++
Tab Width: 8
Ln 19, Col 35
INS