

Practical Test

20% of Module Mark

Fundraising Implementation

PART A - 75%

Children Charity wants to do a charity fund-raising. They would need a distributed system to help them do the fundraising. Children Charity needs your help and kindly asks you to develop a distributed system for their fundraising. Using the application users would be able to donate money as well as to inquire about the amount of money being raised.

A user can donate by specifying the amount of money he/she wants to donate.

Please help Children Charity by building for them a distributed system. The distributed system will be part of your continuous assessment at Networking Programming and Distributed Systems.

In order to build the distributed system you should:

- A. Write an idl consisting of the appropriate operations and attributes for the following services:
 1. Donate money
 2. Get total amount raised
 3. Get number of donors
 4. Returns average amount of money per donor
- 10) marks)
- B. Write a servant that can handle the four services.
- 20) marks)
- C. Write a server process that instantiates the above servant and waits for requests from a client.
- 10) marks)
- D. Write a client that provides a menu which allows making donations, enquiries about the amount of money was raised, number of donors and amount of money per donor.
 1. For the donation option, the client will provide the amount of money that is donated.
 2. For the get total amount raised option, the client will receive the amount of money that was raised and the total amount is printed on the screen.
 3. For the get number of donors, the number is printed on the screen
 4. For the average amount of money per donor, the average is printed on the screen

(30
marks)

E. Demonstrate a working application.

(15 marks)

Note: You have to comment your code for A-D.

PART B - 25%

Describe in details what are the steps needed to follow in order to create the Children Charity fundraising distributed program. What files were generated for you? Provide details of the generated files.

**(25
marks)**

1. First of all, I created a file donation.idl specifying what my system are able to do. Then, I ran idlj to generate my java files. The compiler idlj created six files for me. The _CharityImplBase.java is the skeleton for server, the _CharityStub.java is the stub for client, CharityOperations.java contain all methods of my idl file. CharityHolder.java has operations for arguments. CharityHelper.java has auxiliar functions and Charity.java provide CORBA functionalities. So, I implemented and compiled the Servant, Server and Client java code. This way, starting the server and running the client, the console will ask the user for a operation.