2013

Saket Kumar Pathak
Software Developer 3D Graphics

[Programming in C++ of College days]

Programs are tested with compiler and respective IDEs Bloodshed-DevC++, Visual Studio 2008, Qt 4.2. These are running successfully with console in windows platform. So just enjoy coding. For Visual Studio 2008 and Qt 4.2, please notice the note at the end.

1. **Write a C++ program WAP to print:**

   a) **0**
   1 1
   2 3 5
   8 13 21 34

Program:

```
#include <iostream>

class Fibonacci_Ladder
{
      public:
             Fibonacci_Ladder();
             virtual ~Fibonacci_Ladder();
             int cal_sumation(int i_val);
             void calc_fibonacci_no(int i_count);
             int disp_ladder(int i_nam);

             int *iptr_fib_buffer;

};

Fibonacci_Ladder::Fibonacci_Ladder()
{
}

Fibonacci_Ladder::~Fibonacci_Ladder()
{
}

int Fibonacci_Ladder::cal_sumation(int i_val)
{
    int i_count = 0;
    for (int i_count_1 = 0; i_count_1 <= i_val;
    ++i_count_1)
        for (int i_count_2 = 0; i_count_2 < i_count_1;
    ++i_count_2)
            ++i_count;

    return i_count;
}

void Fibonacci_Ladder::calc_fibonacci_no(int i_count)
{
     int i_count_fib = cal_sumation(i_count);
     iptr_fib_buffer = (int*)malloc(sizeof(int) *
    i_count_fib);

     int i_nam_0, i_nam_1, i_nam_2;
     i_nam_0 = 0;
```

```
        i_nam_1 = 1;

        iptr_fib_buffer[0] = i_nam_0;
        iptr_fib_buffer[1] = i_nam_1;

        for (int i_lp_count = 2; i_lp_count < i_count_fib;
       ++i_lp_count)
        {
            i_nam_2 = i_nam_1 + i_nam_0;
            i_nam_0 = i_nam_1;
            i_nam_1 = i_nam_2;

            iptr_fib_buffer[i_lp_count] = i_nam_2;
        }

        std::cout<<std::endl;
    }

    int Fibonacci_Ladder::disp_ladder(int i_nam)
    {
        calc_fibonacci_no(i_nam);

        int i_size = 0;
        for (int i_r_count = 0; i_r_count <= i_nam;
        ++i_r_count)
        {
            std::cout<<std::endl;
            for (int i_c_count = 0; i_c_count < i_r_count;
        ++i_c_count)
            {
                std::cout<<iptr_fib_buffer[i_size]<<"\t";
                ++i_size;
            }

        }

        std::cout<<std::endl;

        return 0;
    }

    int main()
    {
        int i_choice = 0;
        std::cout<<"\n\n\tPlease enter number of Rows to
        calculate Fibbonacci Ladder: ";
        std::cin>>i_choice;

        Fibonacci_Ladder cl_fib;
        cl_fib.disp_ladder(i_choice);
```

```
        std::cout<<"\n\n\n";
        std::system("pause");
        return 0;
    }
```

2. **Write a C++ program by using functions**
   a) **Sum of Digits of a number (recursive)**
   b) **Maximum digit of a number.**
   c) **Deletion of a specific digit from a number.**

Program:

```
#include <iostream>

class Digit
{
     public:
            Digit();
            virtual ~Digit();

            void set_num(void);
            int get_num(void);

            void sum_of_digit(int i_val = 0);
            void maximum_digit(int i_val = 0);
            void delete_digit(int i_val = 0);

            int i_num;
};

Digit::Digit():
i_num(0)
{
}

Digit::~Digit()
{
}

void Digit::set_num(void)
{
    int i_val = 0;
    std::cout<<"\n\tEnter a number: ";
    std::cin>>i_val;

    i_num = i_val;
}
```

```
int Digit::get_num(void)
{
    return i_num;
}

void Digit::sum_of_digit(int i_val)
{
     static int i_rem = 0;
     i_rem += i_num%10;
     i_num = i_num/10;

     if (i_num > 0)
     {
        sum_of_digit(i_num);
     }
     else
     {
         std::cout<<"\n\tThe sum of Digit:
"<<i_rem<<std::endl;
     }
}

void Digit::maximum_digit(int i_val)
{
    static int i_max = 0;

    int i_rem = i_num%10;
    i_num = i_num/10;

    if (i_max < i_rem)
       i_max = i_rem;

    if (i_num > 0)
    {
          maximum_digit(i_num);
    }
    else
    {
        std::cout<<"The Maximum Digit: "<<i_max;
    }
}

void Digit::delete_digit(int i_val)
{
     int i_rem, i_num_1;
     i_rem = i_num_1 = 0;
     while (i_num > 0)
     {
           i_rem = i_num%10;
           i_num = i_num/10;
```

```
            if (i_rem == i_val)
               std::cout<<"\n\tDigit to delete:
"<<i_rem<<std::endl;
            else
            {
                i_num_1 = i_num_1 * 10 + i_rem;
            }
     }

     while (i_num_1 > 0)
     {
           i_rem = i_num_1%10;
           i_num_1 = i_num_1/10;
           i_num = i_num * 10 + i_rem;
     }

     std::cout<<"\n\tThe number obtained: "<<i_num<<std::endl;
}

int create_menu(void)
{
    std::cout<<"\t\t***Digit Analizer***"<<std::endl;
    std::cout<<"\tTo find sum of digits (Press - 1):
"<<std::endl;
    std::cout<<"\tTo find maximum digit (Press - 2):
"<<std::endl;
    std::cout<<"\tTo delete a digit (Press - 3): "<<std::endl;

    int i_choice = 0;
    std::cout<<"\n\n\tPlease enter your choice: ";
    std::cin>>i_choice;

    return i_choice;
}

int main()
{
    int i_switch = create_menu();

    Digit digit_instance;
    digit_instance.set_num();

    switch (i_switch)
    {
           case 1:
                digit_instance.sum_of_digit();
                break;
           case 2:
                digit_instance.maximum_digit();
                break;
           case 3:
```

```
                int i_digit = 0;
                std::cout<<"\n\tPlease Enter the digit to
delete: ";
                std::cin>>i_digit;
                digit_instance.delete_digit(i_digit);
                break;
    }


    std::cout<<"\n\n\n";
    system("PAUSE");
    return 0;
}
```

3. **Write a C++ program to implement function overloading by implementing the power() in the following different forms . power() to raise a number a^b . The function takes a DOUBLE value for a and INT value for b. Use a default value of 2 for b to make the function to calculate squares when this argument is omitted. And also implements another power function that performs same operations but takes INT for both a and b.**

Program:

```
#include <iostream>

class Power_Calculator
{
      public:
             Power_Calculator();
             virtual ~Power_Calculator();

             double power(double d_base, int i_exponent = 2);
             int power(int i_base, int i_exponent);
};

Power_Calculator::Power_Calculator()
{
}

Power_Calculator::~Power_Calculator()
{
}

double Power_Calculator::power(double d_base, int i_exponent)
{
    int i_count = 0;
    double d_res = 1.0;
    for (i_count = 0; i_count < i_exponent; ++i_count)
```

```
        d_res *= d_base;

    return d_res;
}

int Power_Calculator::power(int i_base, int i_exponent)
{
    int i_count, i_res;
    i_count = i_res = 1;
    for (i_count = 0; i_count < i_exponent; ++i_count)
        i_res *= i_base;

    return i_res;
}

int create_menu(void)
{
    std::cout<<"\t\t***Power Calculator***"<<std::endl;
    std::cout<<"\tTo calculate square of any number
'(double)a' (Press - 1): "<<std::endl;
    std::cout<<"\tTo calculate power of any number 'in form
(int)a raise to (int)b' (Press - 2): "<<std::endl;

    int i_choice = 0;
    std::cout<<"\n\n\tPlease enter your choice: ";
    std::cin>>i_choice;

    return i_choice;
}

int main()
{
    double d_base, d_res;
    d_base = d_res = 0.0;
    int i_res, i_base, i_exponent;
    i_base = i_exponent = i_res = 0;

    int i_switch = create_menu();

    Power_Calculator pow_calc;

    switch (i_switch)
    {
           case 1:
                std::cout<<"\n\tPlease enter base value
(double): ";
                std::cin>>d_base;

                d_res = pow_calc.power(d_base);
                std::cout<<"\n\tThe result: "<<d_res;
                break;
```

```
            case 2:
                 std::cout<<"\n\tPlease enter base value (int):
";
                 std::cin>>i_base;
                 std::cout<<"\n\tPlease enter exponent value
(int): ";
                 std::cin>>i_exponent;

                 i_res = pow_calc.power(i_base, i_exponent);
                 std::cout<<"\n\tThe result: "<<i_res;
                 break;
    }


    std::cout<<"\n\n\n";
    system("PAUSE");
    return 0;
}
```

4. **Create a class called Employee that contains Employee number, employee name, designation, basic pay, deductions (LIC, PF). Include a member function to get data from user for 'n' employees. Write a C++ program to prepare the payslips for 'n' number of employees using the following details :**
   D.A = 40% of basic pay
   H.R.A = 25% of basic pay
   Gpay = basic pay + D.A. + H.R.A.
   Npay = Gpay – deductions
   The result of problems is in given format :
   Emp. no Emp. name Basic D.A. H.R.A. L.I.C. P.F. Gpay Npay

Program:

```
#include <iostream>

class Employee
{
      public:
             Employee();
             virtual ~Employee();

             void get_user_data(void);
             void calc_salary_details(void);
             void display_details(void);

             std::string str_employee_name;
             std::string str_designation;
             int i_employee_number;
             int i_basic_pay;
             int i_deductions_lic;
```

```
            int i_deductions_pf;
            int i_da;
            int i_hra;
            int i_gpay;
            int i_npay;
};

Employee::Employee() :
str_employee_name(""),
str_designation(""),
i_employee_number(0),
i_basic_pay(0),
i_deductions_lic(0),
i_deductions_pf(0)
{
}

Employee::~Employee()
{
}

void Employee::get_user_data()
{
     std::cout<<"\n\n\tEmployee Name: ";
     std::cin>>str_employee_name;
     std::cout<<"\n\tEmployee Number: ";
     std::cin>>i_employee_number;
     std::cout<<"\n\tBasic Pay: ";
     std::cin>>i_basic_pay;
     std::cout<<"\n\tLIC: ";
     std::cin>>i_deductions_lic;
     std::cout<<"\n\tPF: ";
     std::cin>>i_deductions_pf;
     std::cout<<"\n\tDesignation: ";
     std::cin>>str_designation;

     calc_salary_details();
}

void Employee::calc_salary_details()
{
     i_da = 0.4f * i_basic_pay;
     i_hra = 0.25f * i_basic_pay;
     i_gpay = i_basic_pay + i_da + i_hra;
     i_npay = i_gpay - (i_deductions_lic + i_deductions_pf);
}

void Employee::display_details()
{
     std::cout<<"\n\n\tEmployee Name: "<<str_employee_name;
     std::cout<<"\n\tDesignation: "<<str_designation;
```

```
     std::cout<<"\n\tEmployee Number: "<<i_employee_number;
     std::cout<<"\n\tBasic Pay: "<<i_basic_pay;
     std::cout<<"\n\tLIC: "<<i_deductions_lic;
     std::cout<<"\n\tPF: "<<i_deductions_pf;
     std::cout<<"\n\tDA: "<<i_da;
     std::cout<<"\n\tHRA: "<<i_hra;
     std::cout<<"\n\tGPay: "<<i_gpay;
     std::cout<<"\n\tNPay: "<<i_npay;
}

int create_menu(void)
{
    int i_choice;

    std::cout<<"\t\t***Pay Slip Calculator***"<<std::endl;
    std::cout<<"\n\n\tPlease enter number of employees: ";
    std::cin>>i_choice;

    return i_choice;
}

int main()
{
    int i_num = create_menu();
    Employee *emp = (Employee*)malloc(sizeof(Employee) *
i_num);

    for (int i_count = 0; i_count < i_num; ++i_count)
    {
        emp[i_count].get_user_data();
    }

    for (int i_count = 0; i_count < i_num; ++i_count)
    {
        emp[i_count].display_details();
    }

    std::cout<<"\n\n\n";
    system("PAUSE");
    return 0;
}
```

5. **Create a class ElectricityBill that contains Customer _number, Customer_name, Customer _age, Customer_address, unit_consumed,Customer_bill. Write a Menu driven C++ program to display the Customer _number, Customer_name and Customer_address of the Customer for the following conditions:**
   (i) if Unit_Consumed is less than equal to 200 then rate is 4 Rs/Unit.
   (ii) if Unit_Consumed is greater than 200 and less than equal to 500 then rate is 5 Rs/Units for units exceeding units 200.

(iii) if Unit_Consumed is greater than 500 then rate is 6 Rs/Units, first 200 Units rate will be again 4 Rs/Unit and 200 to 500 units will be charged by 5 Rs/Units.

Program:

```
#include <iostream>
#include <cstdlib>

class ElectricityBill_Generator
{
      public:
             ElectricityBill_Generator();
             virtual ~ElectricityBill_Generator();

             void set_unit_consumed(int il_units_consumed);
             void set_details(int il_units_consumed);
             void display_details();

             int i_customer_number;
             int i_customer_age;
             int i_units_consumed;
             std::string str_customer_address;
             std::string str_customer_name;
             double d_customer_bill;
};

ElectricityBill_Generator::ElectricityBill_Generator()
{
     i_customer_number = 0;
     i_customer_age = 0;
     i_units_consumed = 0;
     str_customer_address = "";
     str_customer_name = "";
     d_customer_bill = 0.0;
}

ElectricityBill_Generator::~ElectricityBill_Generator()
{
}

void ElectricityBill_Generator::set_unit_consumed(int
il_units_consumed)
{
     i_units_consumed = il_units_consumed;
     set_details(i_units_consumed);
}

void ElectricityBill_Generator::set_details(int
il_units_consumed)
{
```

```
     double d_rate = 0.0;
     std::string str_temp = "";

     if (il_units_consumed < 200)
     {
        d_rate = 4.0;

        d_customer_bill = il_units_consumed * d_rate;
        i_customer_number = rand() * 1000;
        i_customer_age = (rand() * 100) - 30;

        for (int i_count = 0; i_count < 4; ++i_count)
            str_temp += ((rand() * 10) + 'a');
        str_customer_name = str_temp;

        str_temp = "";
        for (int i_count = 0; i_count < 13; ++i_count)
            str_temp += ((rand() * 10) + 'a');
        str_customer_address = str_temp;
     }
     else if ((il_units_consumed > 200)&&(il_units_consumed <
500))
     {
          d_rate = 5.0;
          int i_temp_units = 0;

          i_temp_units = (il_units_consumed - 200);
          d_customer_bill = i_temp_units * (d_rate);
          d_customer_bill += (il_units_consumed -
i_temp_units) * (d_rate - 1);

          i_customer_number = rand() % 1000;
          i_customer_age = (rand() % 100) - 30;

          for (int i_count = 0; i_count < 4; ++i_count)
              str_temp += ((rand() % 10) + 'a');
          str_customer_name = str_temp;

          str_temp = "";
          for (int i_count = 0; i_count < 13; ++i_count)
              str_temp += ((rand() % 10) + 'a');
          str_customer_address = str_temp;
     }
     else if (il_units_consumed > 500)
     {
          d_rate = 6.0;
          int i_temp_units = 0;

          d_customer_bill += 200 * (d_rate - 2);
          i_temp_units = (il_units_consumed - 200);
          d_customer_bill += 300 * (d_rate - 1);
```

```
            i_temp_units = (il_units_consumed - 300);
            d_customer_bill += i_temp_units * (d_rate);

            i_customer_number = rand() % 1000;
            i_customer_age = (rand() % 100) - 30;

            for (int i_count = 0; i_count < 4; ++i_count)
                str_temp += ((rand() % 10) + 'a');
            str_customer_name = str_temp;

            str_temp = "";
            for (int i_count = 0; i_count < 13; ++i_count)
                str_temp += ((rand() % 10) + 'a');
            str_customer_address = str_temp;
        }

        display_details();
}

void ElectricityBill_Generator::display_details()
{
        std::cout<<"\tCustomer Number:
"<<i_customer_number<<std::endl;
        std::cout<<"\tAge: "<<i_customer_age<<std::endl;
        std::cout<<"\tUnits Consumed:
"<<i_units_consumed<<std::endl;
        std::cout<<"\tBill: "<<d_customer_bill<<std::endl;
        std::cout<<"\tName of Customer:
"<<str_customer_name<<std::endl;
        std::cout<<"\tAddress:
"<<str_customer_address<<std::endl;
        std::cout<<"Sorry it's your Address ;)";
}

int get_units(void)
{
    int i_choice = 0;
    std::cout<<"\t\t***Electricity Bill
Generator***"<<std::endl;
    std::cout<<"\n\n\tPlease Enter units consumed: ";
    std::cin>>i_choice;

    std::cout<<std::endl;

    return i_choice;
}

int main()
{
    int i_units = get_units();
```

```
    ElectricityBill_Generator bill_gen;
    bill_gen.set_unit_consumed(i_units);

    std::cout<<"\n\n\n";
    system("PAUSE");
    return 0;
}
```

6. **Declare a class called time having hours, minutes and seconds as member variables. Define a member function called add_time that accepts two objects of type time class, performs addition of time in hours, minutes and seconds format and returns an object of type time class. Write appropriate functions for initialization and display of member functions.**

Program:

```
#include <iostream>

class Time
{
      public:
             Time();
             virtual ~Time();

             void set_time(void);
             void get_time(void);

             int i_hours;
             int i_mins;
             int i_secs;
};

Time::Time():
i_hours(0),
i_mins(0),
i_secs(0)
{
}

Time::~Time()
{
}

void Time::set_time()
{
     std::cout<<"\n\n\tHours: ";
     std::cin>>i_hours;
     std::cout<<"\n\tMinutes: ";
     std::cin>>i_mins;
     std::cout<<"\n\tSeconds: ";
```

```
     std::cin>>i_secs;
}

void Time::get_time()
{
     std::cout<<"\n\n\tHours: "<<i_hours;
     std::cout<<"\n\tMinutes: "<<i_mins;
     std::cout<<"\n\tSeconds: "<<i_secs;
}

Time& add_time(Time& t_ob_1, Time& t_ob_2)
{
      static Time t_res;
      int i_temp;

      if (t_ob_1.i_hours > 24)
         t_ob_1.i_hours /= 24;

      if (t_ob_1.i_mins >= 60)
      {
         i_temp = t_ob_1.i_mins / 60;
         t_ob_1.i_hours += i_temp;

         t_ob_1.i_mins = t_ob_1.i_mins % 60;
      }

      if (t_ob_1.i_secs >= 60)
      {
         i_temp = t_ob_1.i_secs / 60;
         t_ob_1.i_mins += i_temp;

         t_ob_1.i_secs = t_ob_1.i_secs % 60;
      }

      if (t_ob_2.i_hours > 24)
         t_ob_2.i_hours /= 24;

      if (t_ob_2.i_mins >= 60)
      {
         i_temp = t_ob_2.i_mins / 60;
         t_ob_2.i_hours += i_temp;

         t_ob_2.i_mins = t_ob_2.i_mins % 60;
      }

      if (t_ob_2.i_secs >= 60)
      {
         i_temp = t_ob_2.i_secs / 60;
         t_ob_2.i_mins += i_temp;

         t_ob_2.i_secs = t_ob_2.i_secs % 60;
```

```
	}

	t_res.i_hours = t_ob_1.i_hours + t_ob_2.i_hours;
	t_res.i_mins = t_ob_1.i_mins + t_ob_2.i_mins;
	t_res.i_secs = t_ob_1.i_secs + t_ob_2.i_secs;

	return t_res;
}

int create_menu(void)
{
    std::cout<<"\t\t***Time Calculator***\n\n"<<std::endl;
}

int main()
{
    create_menu();

    Time t_obj_1;
    t_obj_1.set_time();

    Time t_obj_2;
    t_obj_2.set_time();

    std::cout<<"\n\n\tAdded Result of Time: "<<std::endl;

    Time time = add_time(t_obj_1, t_obj_2);
    time.get_time();

    std::cout<<"\n\n\n";
    system("PAUSE");
    return 0;
}
```

7. **Write C++ programs that illustrate various types of constructors.**

Program:

```
#include <iostream>

class Employee
{
	public:
		Employee();
		virtual ~Employee();

		virtual void display(void);

		std::string str_emp_name;
		std::string str_emp_code;
```

```
            int i_salary;
};

Employee::Employee()
{
    str_emp_name = "__blank__";
    str_emp_code = "__blank__";
    i_salary = 0;
}

Employee::~Employee()
{
}

void Employee::display()
{
     std::cout<<"\n\n\tEmployee Name: "<<str_emp_name;
     std::cout<<"\n\tEmployee Code: "<<str_emp_code;
     std::cout<<"\n\tEmployee Salary: "<<i_salary;
}

class Manager : public Employee
{
      public:
             Manager();
             virtual ~Manager();

             void set_value(void);
             void display(void);
             int i_num_assignments;
};

Manager::Manager()
{
}

Manager::~Manager()
{
}

void Manager::set_value(void)
{
     std::cout<<"\n\tPlease enter name: ";
     std::cin>>str_emp_name;
     std::cout<<"\n\tPlease enter code: ";
     std::cin>>str_emp_code;
     std::cout<<"\n\tPlease eneter salary: ";
     std::cin>>i_salary;
     std::cout<<"\n\tPlease eneter numnber of assignment: ";
     std::cin>>i_num_assignments;
}
```

```
void Manager::display(void)
{
     std::cout<<"\n\n\tEmployee (Manager) Details you Entered
"<<std::endl;
     std::cout<<"\n\tEmployee Name: "<<str_emp_name;
     std::cout<<"\n\tEmployee Code: "<<str_emp_code;
     std::cout<<"\n\tEmployee Salary: "<<i_salary;
     std::cout<<"\n\tNumber of Assignment:
"<<i_num_assignments;
}

class Clerk : public Employee
{
      public:
             Clerk();
             virtual ~Clerk();

             void set_value(void);
             void display(void);

             int i_allowance;
};

Clerk::Clerk()
{
}

Clerk::~Clerk()
{
}

void Clerk::set_value(void)
{
     std::cout<<"\n\tPlease enter name: ";
     std::cin>>str_emp_name;
     std::cout<<"\n\tPlease enter code: ";
     std::cin>>str_emp_code;
     std::cout<<"\n\tPlease eneter salary: ";
     std::cin>>i_salary;
     std::cout<<"\n\tPlease eneter allowance: ";
     std::cin>>i_allowance;
}

void Clerk::display(void)
{
     std::cout<<"\n\n\tEmployee (Clerk) Details you Entered
"<<std::endl;
     std::cout<<"\n\tEmployee Name: "<<str_emp_name;
     std::cout<<"\n\tEmployee Code: "<<str_emp_code;
     std::cout<<"\n\tEmployee Salary: "<<i_salary;
```

```
     std::cout<<"\n\tAllowance: "<<i_allowance;
}

int create_menu(void)
{
    std::cout<<"\t\t***Employee (Inheritance
Relation)***\n\n"<<std::endl;

    std::cout<<"\tTo enter details for Manager (Press - 1):
"<<std::endl;
    std::cout<<"\tTo enter details for Clerk (Press - 2):
"<<std::endl;


    int i_choice = 0;
    std::cout<<"\n\n\tPlease enter your choice: ";
    std::cin>>i_choice;

    return i_choice;
}

int main()
{
    int i_switch = create_menu();

    switch (i_switch)
    {
           case 1:
                {
                  Manager obj_manager;
                  obj_manager.set_value();
                  obj_manager.display();
                  break;
                }
           case 2:
                {
                  Clerk obj_clerk;
                  obj_clerk.set_value();
                  obj_clerk.display();
                  break;
                }
    }

    std::cout<<"\n\n\n";
    system("PAUSE");
    return 0;
}
```

8. **Write C++ programs that illustrate how the following forms of inheritance are supported:**
   a) Single inheritance

b) Multiple inheritance
c) Multi level inheritance
d) Hierarchical inheritance

Program:

```
#include <iostream>

class Human_Being
{
	public:
		Human_Being();
		virtual ~Human_Being();

		virtual void human(int i_age, std::string
str_gender) = 0;

		int i_age;
		std::string str_gender;
};

Human_Being::Human_Being()
{
}

Human_Being::~Human_Being()
{
}

class Girl : public Human_Being
{
	public:
		Girl();
		virtual ~Girl();

		void human(int il_age, std::string strl_gender);

		int i_age;
		std::string str_gender;
};

Girl::Girl()
{
}

Girl::~Girl()
{
}

void Girl::human(int il_age, std::string strl_gender)
{
```

```
     i_age = il_age;
     str_gender = strl_gender;

     std::cout<<"\n\n\tYou Entered "<<std::endl;
     std::cout<<"\n\tAge: "<<i_age;
     std::cout<<"\n\tGender: "<<str_gender;
}

class Boy : public Human_Being
{
      public:
             Boy();
             virtual ~Boy();

             void human(int il_age, std::string strl_gender);

             int i_age;
             std::string str_gender;
};

Boy::Boy()
{
}

Boy::~Boy()
{
}

void Boy::human(int il_age, std::string strl_gender)
{
     i_age = il_age;
     str_gender = strl_gender;

     std::cout<<"\n\n\tYou Entered "<<std::endl;
     std::cout<<"\n\tAge: "<<i_age;
     std::cout<<"\n\tGender: "<<str_gender;
}

int create_menu(void)
{
    std::cout<<"\t\t***Humanity Test***\n\n"<<std::endl;

    std::cout<<"\tTo enter details for Boy (Press - b):
"<<std::endl;
    std::cout<<"\tTo enter details for Girl (Press - g):
"<<std::endl;

    char ch_choice;
    std::cout<<"\n\n\tPlease enter your choice: ";
    std::cin>>ch_choice;
```

```
    return ch_choice;
}

int main()
{
    int i_switch = create_menu();

    int i_age;
    std::string str_gender;
    switch (i_switch)
    {
           case 'b':
                {
                  std::cout<<"\n\tPlease Age: ";
                  std::cin>>i_age;

                  Boy obj_b;
                  obj_b.human(i_age, "Boy");
                  break;
                }
           case 'g':
                {
                  std::cout<<"\n\tPlease Age: ";
                  std::cin>>i_age;

                  Girl obj_g;
                  obj_g.human(i_age, "Girl");

                  break;
                }
    }

    std::cout<<"\n\n\n";
    system("PAUSE");
    return 0;
}
```

9. Write a C++ program that illustrates the role of pure virtual function.

Program:

```
#include <iostream>

class Human_Being
{
      public:
             Human_Being();
             virtual ~Human_Being();

             virtual void human(int i_age, std::string
str_gender) = 0;
```

```
              int i_age;
              std::string str_gender;
};

Human_Being::Human_Being()
{
}

Human_Being::~Human_Being()
{
}

class Girl : public Human_Being
{
      public:
              Girl();
              virtual ~Girl();

              void human(int il_age, std::string strl_gender);

              int i_age;
              std::string str_gender;
};

Girl::Girl()
{
}

Girl::~Girl()
{
}

void Girl::human(int il_age, std::string strl_gender)
{
     i_age = il_age;
     str_gender = strl_gender;

     std::cout<<"\n\n\tYou Entered "<<std::endl;
     std::cout<<"\n\tAge: "<<i_age;
     std::cout<<"\n\tGender: "<<str_gender;
}

class Boy : public Human_Being
{
      public:
              Boy();
              virtual ~Boy();

              void human(int il_age, std::string strl_gender);
```

```
            int i_age;
            std::string str_gender;
};

Boy::Boy()
{
}

Boy::~Boy()
{
}

void Boy::human(int il_age, std::string strl_gender)
{
     i_age = il_age;
     str_gender = strl_gender;

     std::cout<<"\n\n\tYou Entered "<<std::endl;
     std::cout<<"\n\tAge: "<<i_age;
     std::cout<<"\n\tGender: "<<str_gender;
}

int create_menu(void)
{
    std::cout<<"\t\t***Humanity Test***\n\n"<<std::endl;

    std::cout<<"\tTo enter details for Boy (Press - b):
"<<std::endl;
    std::cout<<"\tTo enter details for Girl (Press - g):
"<<std::endl;

    char ch_choice;
    std::cout<<"\n\n\tPlease enter your choice: ";
    std::cin>>ch_choice;

    return ch_choice;
}

int main()
{
    int i_switch = create_menu();

    int i_age;
    std::string str_gender;
    switch (i_switch)
    {
           case 'b':
                {
                  std::cout<<"\n\tPlease Age: ";
                  std::cin>>i_age;
```

```
                    Boy obj_b;
                    obj_b.human(i_age, "Boy");
                    break;
                  }
            case 'g':
                  {
                    std::cout<<"\n\tPlease Age: ";
                    std::cin>>i_age;

                    Girl obj_g;
                    obj_g.human(i_age, "Girl");

                    break;
                  }
    }

    std::cout<<"\n\n\n";
    system("PAUSE");
    return 0;
}
```

10. **Create a class employee which has name, employee code and salary as its data members. Derive two classes called manager and clerk from employee. Manager has number of assistance as its data members and clerk has allowance as its data members. Override the function display() in both the derived classes which display information related to it. Use constructor and show how to achieve run time polymorphism.**

Program:

```
#include <iostream>

class Constructor_Types
{
      public:
             Constructor_Types();
             virtual ~Constructor_Types();

             Constructor_Types(int il_data_mem);
             Constructor_Types(const Constructor_Types
&obj_ctor);

             int i_data_mem;
};

Constructor_Types::Constructor_Types():
i_data_mem(0)
{
   std::cout<<"\n\tDefault Constructor"<<std::endl;
   std::cout<<"\tDefault value: "<<i_data_mem<<std::endl;
}
```

```
Constructor_Types::~Constructor_Types()
{
}

Constructor_Types::Constructor_Types(int il_data_mem)
{
   i_data_mem = il_data_mem;

   std::cout<<"\n\tParametrized Constructor"<<std::endl;
   std::cout<<"\tStored value: "<<i_data_mem<<std::endl;
}

Constructor_Types::Constructor_Types(const Constructor_Types
&obj_ctor)
{
   i_data_mem = obj_ctor.i_data_mem;

   std::cout<<"\n\tCopy Constructor"<<std::endl;
   std::cout<<"\n\tStored value: "<<i_data_mem<<std::endl;
}


int create_menu(void)
{
    std::cout<<"\n\t***Employee (Inheritance
Relation)***\n\n"<<std::endl;

    std::cout<<"\n\tFor Default C-tor (Press - 1):
"<<std::endl;
    std::cout<<"\n\tFor Parametrized C-tor (Press - 2):
"<<std::endl;
    std::cout<<"\n\tFor Copy C-tor (Press - 3): "<<std::endl;

    int i_choice = 0;
    std::cout<<"\n\n\tPlease enter your choice: ";
    std::cin>>i_choice;

    return i_choice;
}

int main()
{
    int i_switch = create_menu();

    switch (i_switch)
    {
           case 1:
                {
                  std::cout<<"\n\tDefault C-tor"<<std::endl;
                  Constructor_Types obj_ctor;
```

```
                    std::cout<<"\n\t----------------
\n"<<std::endl;
                    break;
                }
            case 2:
                {
                    std::cout<<"\n\tParameterized C-
tor"<<std::endl;
                    Constructor_Types obj_ctor(10);
                    std::cout<<"\n\t----------------
\n"<<std::endl;
                    break;
                }
            case 3:
                {
                    std::cout<<"\n\tCopy C-tor"<<std::endl;
                    Constructor_Types obj_ctor(10);
                    Constructor_Types obj_cpy_ctor(obj_ctor);
                    std::cout<<"\n\t----------------
\n"<<std::endl;
                    break;
                }
    }

    std::cout<<"\n\n\n";
    system("PAUSE");
    return 0;
}
```

11. Write a C++ program to that counts the characters, lines and words in the text file.

Program:

```
#include <iostream>

template <class hi>hi gap(hi frnd_1, hi frnd_2)
{
        return (frnd_1 < frnd_2 ? frnd_2 : frnd_1);
}


template <class hi, class bye>hi long_gap(hi temp_1, bye
temp_2)
{
        return (temp_1 < temp_2 ? temp_2 : temp_1);
}
```

```
int main()
{
      int inam_1, inam_2, iresult;
      long lnam_1, lnam_2, lresult;

      inam_1 = 3;
      inam_2 = inam_1++;

      iresult = gap<int>(inam_1, inam_2);
      printf("Greater Number(int): %d\n", iresult);
     lnam_1 = 1020304050;
      lnam_2 = lnam_1++;

      lresult = long_gap<long>(lnam_1, lnam_2);
      printf("Greater Number(long): %ld\n", lresult);

      lresult = long_gap<int,long>(inam_1, lnam_2);
      printf("Greater Number(long): %ld", lresult);

      getchar();
      return 0;
}
```

12. Write a C++ program which copies one file to another.

Program:

```
#include <iostream>
#include <fstream>
#include <string>

class File_Copy
{
       public:
              File_Copy();
              virtual ~File_Copy();

              void create_file(void);
              void read_write_data(void);
};

File_Copy::File_Copy()
{
}
```

```
File_Copy::~File_Copy()
{
}

void File_Copy::create_file()
{
   std::fstream l_txt_file;
   l_txt_file.open("hi_file.txt", std::ios::out);
   l_txt_file<<"This is Saket. \nI like C++.\nThat's it
... ;)\n";

   l_txt_file.close();
}

void File_Copy::read_write_data()
{
     std::fstream read_txt_file;
     read_txt_file.open("hi_file.txt", std::ios::in);
     std::cout<<"\n\n\tReading From File:
hi_file.txt"<<std::endl;

     std::fstream write_txt_file;
     write_txt_file.open("bye_file.txt", std::ios::out);
     std::cout<<"\n\n\tWriting To File:
bye_file.txt"<<std::endl;

     while (!read_txt_file.eof())
     {
           char temp_char = read_txt_file.get();
           write_txt_file<<temp_char;
     }

     write_txt_file.close();
     read_txt_file.close();
}

void create_menu(void)
{
    std::cout<<"\n\t***File Copy***\n\n"<<std::endl;
}

int main()
{
```

```
        create_menu();

        File_Copy obj_file;
        obj_file.create_file();

        obj_file.read_write_data();

        std::cout<<"\n\n\n";
        system("PAUSE");
        return 0;
}
```

13. Write a C++ program to find a word in the text file.

Program:

```
#include <iostream>
#include <fstream>
#include <string>

class File_Search
{
      public:
             File_Search();
             virtual ~File_Search();

             void create_file(void);
             void search_word(void);
};

File_Search::File_Search()
{
}

File_Search::~File_Search()
{
}

void File_Search::create_file(void)
{
   std::fstream l_txt_file;
   l_txt_file.open("search_file.txt", std::ios::out);
   l_txt_file<<"This is Saket. \nI like C++.\nThat's it
... ;)\n";
```

```
   l_txt_file.close();
}

void File_Search::search_word(void)
{
     std::fstream read_txt_file;
     read_txt_file.open("search_file.txt", std::ios::in);
     std::cout<<"\n\n\tSearching From File:
search_file.txt"<<std::endl;

     std::string str_word;
     std::cout<<"\n\tPlease enter word to search: ";
     std::cin>>str_word;

     int i_check = 0;
     while (!read_txt_file.eof())
     {
           std::string str_temp = "";
           char ch_temp = 'a';
           if ((ch_temp = read_txt_file.get()) != ' ')
              str_temp += ch_temp;
           else
           {
               str_temp += '\0';
               if (str_word.compare(str_temp))
               {
                  std::cout<<"\n\tWord Found";
                  i_check = 1;
                  break;
               }
           }
     }

     if (i_check == 0)
        std::cout<<"\n\tWord Not Found";
}

void create_menu(void)
{
    std::cout<<"\n\t***File Search***\n\n"<<std::endl;
}

int main()
```

```
    {
        create_menu();

        File_Search obj_file;
        obj_file.create_file();

        obj_file.search_word();

        std::cout<<"\n\n\n";
        system("PAUSE");
        return 0;
}
```

14. Write a C++ program to implement Stack using Template.

Program:

```
#include <iostream>
#include <stack>
#include <string>

class STL_Stack
{
      public:
             STL_Stack();
             virtual ~STL_Stack();

             void fill_stack(void);
             void disp_stack(void);

             std::stack<std::string> oops_stack;
};

STL_Stack::STL_Stack()
{
}

STL_Stack::~STL_Stack()
{
}

void STL_Stack::fill_stack()
{
```

```
	std::cout<<"\n\n\tEnter 000 to terminate:
"<<std::endl;

	std::string str_temp;
	std::cout<<"\n\tEnter Name: ";
	std::cin>>str_temp;
	oops_stack.push(str_temp);

	while (str_temp.compare("000") != 0)
	{
		std::cout<<"\n\tEnter Name: ";
		std::cin>>str_temp;
		oops_stack.push(str_temp);
	}

	if (str_temp.compare("000") == 0)
	   std::cout<<"\n\tProcess terminated."<<std::endl;
}

void STL_Stack::disp_stack()
{
	std::cout<<"\n\tStack Content: "<<std::endl;
	while (!oops_stack.empty())
	{
		std::cout<<"\n\t"<<
oops_stack.top()<<std::endl;
		oops_stack.pop();
	}

	std::cout<<"\n\tStack Empty."<<std::endl;
}

void create_menu(void)
{
    std::cout<<"\n\t***Stack***\n\n"<<std::endl;
}

int main()
{
    create_menu();

    STL_Stack obj_stack;
    obj_stack.fill_stack();
    obj_stack.disp_stack();
```

```
        std::cout<<"\n\n\n";
        system("PAUSE");
        return 0;
}
```

15. Write a C++ program to implement Queue using Template.

Program:

```
#include <iostream>
#include <queue>
#include <string>

class STL_Queue
{
      public:
             STL_Queue();
             virtual ~STL_Queue();

             void fill_queue(void);
             void disp_queue(void);

             std::queue<std::string> oops_queue;
};

STL_Queue::STL_Queue()
{
}

STL_Queue::~STL_Queue()
{
}

void STL_Queue::fill_queue()
{
     std::cout<<"\n\n\tEnter 000 to terminate:
"<<std::endl;

     std::string str_temp;
     std::cout<<"\n\tEnter Name: ";
     std::cin>>str_temp;
     oops_queue.push(str_temp);
```

```
      while (str_temp.compare("000") != 0)
      {
            std::cout<<"\n\tEnter Name: ";
            std::cin>>str_temp;
            oops_queue.push(str_temp);
      }

      if (str_temp.compare("000") == 0)
         std::cout<<"\n\tProcess terminated."<<std::endl;
}

void STL_Queue::disp_queue()
{
      std::cout<<"\n\tStack Content: "<<std::endl;
      while (!oops_queue.empty())
      {
            std::cout<<"\n\t"<<
oops_queue.front()<<std::endl;
            oops_queue.pop();
      }

      std::cout<<"\n\tStack Empty."<<std::endl;
}

void create_menu(void)
{
    std::cout<<"\n\t***Stack***\n\n"<<std::endl;
}

int main()
{
    create_menu();

    STL_Queue obj_queue;
    obj_queue.fill_queue();
    obj_queue.disp_queue();

    std::cout<<"\n\n\n";
    system("PAUSE");
    return 0;
}
```

16. Write a C++ program to create generic function using Template.

Program:

```
#include <iostream>

template <class hi>hi gap(hi frnd_1, hi frnd_2)
{
         return (frnd_1 < frnd_2 ? frnd_2 : frnd_1);
}


template <class hi, class bye>hi long_gap(hi temp_1, bye
temp_2)
{
         return (temp_1 < temp_2 ? temp_2 : temp_1);
}

int main()
{
     int inam_1, inam_2, iresult;
     long lnam_1, lnam_2, lresult;

     inam_1 = 3;
     inam_2 = inam_1++;

     iresult = gap<int>(inam_1, inam_2);
     printf("Greater Number(int): %d\n", iresult);
     lnam_1 = 1020304050;
     lnam_2 = lnam_1++;

     lresult = long_gap<long>(lnam_1, lnam_2);
     printf("Greater Number(long): %ld\n", lresult);

     lresult = long_gap<int,long>(inam_1, lnam_2);
     printf("Greater Number(long): %ld", lresult);

     getchar();
     return 0;
}
```

Note:

- **For Visual Studio 2008 and Qt (version- 4.2) you need to format the code as, make all the initialization throughout the function to very initial steps after opening braces.**