**Assignment - 31 A Job Ready Bootcamp in C++, DSA and IOT MySirG**

**new and delete operator, Inheritance**

1. Define a class Person with instance members name and age. Also define member

functions setName(), setAge(), getName(), getAge(). Now define class Employee by

inheriting Person class. In the Employee class define empid and salary as instance

members. Also define setEmpid, setSalary, getEmpid, getSalary.

Sol – 1.

#include<iostream>

#include<cstring>

using namespace std;

class Person

{

protected :

char name[20];

int age;

public :

void setName(char \*a)

{

strcpy(name,a);

}

void setAge(int a)

{

age=a;

}

char \* getName()

{

return name;

}

int getAge()

{

return age;

}

};

class Employee : public Person

{

private :

int salary;

int empid;

public :

void setSalary(int a)

{

salary=a;

}

void setEmpid(int a)

{

empid=a;

}

int getSalary()

{

return salary;

}

int getEmpid()

{

return empid;

}

void display()

{

cout<<"Name : "<<name<<" Age : "<<age<<" Empid : "<<empid<<" Salary : "<<salary<<endl;

}

};

int main()

{

Employee E;

E.setName("Gurudev");

E.setAge(18);

E.setEmpid(1);

E.setSalary(100000);

E.display();

return 0;

}

2. Write a C++ program to add two numbers using single inheritance. Accept these two

numbers from the user in base class and display the sum of these two numbers in

derived class.

Sol – 2.

#include<iostream>

#include<cstring>

using namespace std;

class Num

{

protected :

int a,b;

public :

void accept(int x,int y)

{

a=x;

b=y;

}

};

class sums : public Num

{

public :

void printsum()

{

cout<<"Sum is : "<<(a+b)<<endl;

}

};

int main()

{

sums s;

s.accept(5,10);

s.printsum();

return 0;

}

3. Write a C++ program to calculate the percentage of a student using multi-level

inheritance. Accept the marks of three subjects in base class. A class will be derived

from the above mentioned class which includes a function to find the total marks

obtained and another class derived from this class which calculates and displays the

percentage of students.

Sol – 3.

#include<iostream>

#include<cstring>

using namespace std;

class Num

{

protected :

int a,b,c;

public :

void accept()

{

cout<<"Enter marks of 3 subjects : "<<endl;

cin>>a>>b>>c;

}

};

class sum : public Num

{

protected :

int sum;

public :

void total()

{

sum=a+b+c;

}

};

class percentage : public sum

{

public :

void printpercent()

{

cout<<"Student got : "<<(sum/3.0)<<"%"<<endl;

}

};

int main()

{

percentage p;

p.accept();

p.total();

p.printpercent();

return 0;

}

4. Write a C++ program to design a base class Person (name, address,

phone\_no). Derive a class Employee (eno, ename) from Person. Derive a

class Manager (designation, department name, basic-salary) from

Employee. Write a menu driven program to:

a. Accept all details of 'n' managers.

b. Display manager having highest salary

Sol – 4.

#include<iostream>

#include<cstring>

using namespace std;

class Person

{

protected :

int ph;

char name[20];

char add[30];

};

class Employee : public Person

{

protected :

int eno;

char ename[20];

};

class Manager : public Employee

{

private :

int bsal;

char dept[25];

char des[15];

public :

int getbsal()

{

return bsal;

}

char\* getename()

{

return ename;

}

void accept()

{

cout<<"\nEnter Details of Manager\n-------------------------"<<endl;

cout<<"Enter employee No. : ";

cin>>eno;

cout<<"Enter employee Name : ";

cin.get();

fgets(ename,20,stdin);

ename[strlen(ename)-1]=0;

cout<<"Enter Address : ";

fgets(add,30,stdin);

add[strlen(add)-1]=0;

cout<<"Enter Phone No. : ";

cin>>ph;

cout<<"Enter Designation : ";

cin.get();

fgets(des,20,stdin);

des[strlen(des)-1]=0;

cout<<"Enter Department Name : ";

fgets(dept,25,stdin);

dept[strlen(dept)-1]=0;

cout<<"Enter Basic Salary : ";

cin>>bsal;

}

};

int main()

{

int a,i,temp=0;

cout<<"How many Managers you want to enter : ";

cin>>a;

Manager M[a];

for(i=0;i<a;i++)

{

M[i].accept();

}

for(i=0;i<a;i++)

{

if(M[temp].getbsal()<M[i].getbsal())

temp=i;

}

cout<<"\nManager with highest salary is : "<<M[temp].getbsal();

cout<<"\nName : "<<M[temp].getename();

return 0;

}

5. Write a C++ program to define a base class Item (item-no, name, price).

Derive a class Discounted-Item (discount-percent). A customer purchases

'n' items. Display the item-wise bill and total amount using appropriate

format.

Sol – 5.

#include<iostream>

#include<cstring>

using namespace std;

class Item

{

protected :

int itemno,price;

char name[20];

};

class Disitem : public Item

{

private :

float dispercent,disprice;

public :

int getprice()

{

return price;

}

int getdisprice()

{

return price-disprice;

}

void acceptval()

{

cout<<"Enter Item Name : ";

fflush(stdin);

fgets(name,20,stdin);

name[strlen(name)-1]=0;

cout<<"Enter Item No : ";

cin>>itemno;

cout<<"Enter Item Price : ";

cin>>price;

cout<<"Enter Discount Percent : ";

cin>>dispercent;

disprice=price-price\*dispercent/100;

}

void display()

{

cout<<"Item Name : "<<name<<endl;

cout<<"Item No : "<<itemno<<endl;

cout<<"Item Price : "<<price<<endl;

cout<<"Discounted Percent : "<<dispercent<<endl;

cout<<"Discounted Price : "<<disprice<<endl;

}

};

int main()

{

Disitem a[100];

int n,finprice=0,findis=0;

cout<<"How many items you want to enter : ";

cin>>n;

for(int i=0;i<n;i++)

{

a[i].acceptval();

cout<<"\n---------------------\n";

}

for(int i=0;i<n;i++)

{

a[i].display();

cout<<"\n---------------------\n";

}

for(int i=0;i<n;i++)

{

finprice=finprice+a[i].getprice();

findis=findis+a[i].getdisprice();

}

cout<<"Total Price : "<<finprice<<endl;

cout<<"Total Discount : "<<findis;

cout<<"\nBill : "<<finprice-findis;

return 0;

}

6. Write a C++ program to demonstrate how a common friend function can

be used to exchange the private values of two classes. (Use call by

reference method).

Sol – 6.

#include<iostream>

using namespace std;

class B;

class A

{

private :

int a;

public :

A()

{

a=10;

}

void display()

{

cout<<"Value of A : "<<a<<endl;

}

friend void swap(A \*p,B \*q);

};

class B

{

private :

int b;

public :

B(){b=20;}

void display()

{

cout<<"Value of B : "<<b<<endl;

}

friend void swap(A \*p,B \*q);

};

avoid swap(A \*p,B \*q)

{

int c;

c=p->a;

p->a=q->b;

q->b=c;

}

int main()

{

A a;

B b;

swap(&a,&b);

a.display();

b.display();

return 0;

}

7. Write class declarations and member function definitions for a C++ base

class to represent an Employee (emp-code, name).

Derive two classes as Fulltime (daily rate, number of days, salary) and

Parttime (number of working hours, hourly rate, salary).

Write a menu driven program to:

1. Accept the details of ‘n’ employees.

2. Display the details of ‘n’ employees.

3. Search a given Employee by emp-code.

Sol – 7.

#include<iostream>

#include<cstring>

using namespace std;

class Employee

{

public :

int empcode=0;

char name[20];

};

class Fulltime : public Employee

{

public :

int daily\_rate,nod,salary;

void setdata()

{

cout<<"-------------------------"<<endl;

cout<<"Employee ID : ";

cin>>empcode;

cout<<"Employee Name : ";

fflush(stdin);

fgets(name,20,stdin);

name[strlen(name)-1]=0;

cout<<"Employee Daily Rate : ";

cin>>daily\_rate;

cout<<"No of Days : ";

cin>>nod;

cout<<"\nSalary : "<<daily\_rate\*nod<<endl;

cout<<"-------------------------"<<endl;

}

int getempid()

{

return empcode;

}

void show()

{

cout<<"-------------------------"<<endl;

cout<<"Status : Full Time"<<endl;

cout<<"Employee ID : "<<empcode<<endl;

cout<<"Employee Name : "<<name<<endl;

cout<<"Employee Hourly Rate : "<<daily\_rate<<endl;

cout<<"No of Hours : "<<nod<<endl;

cout<<"\nSalary : "<<daily\_rate\*nod<<endl;

cout<<"-------------------------"<<endl;

}

};

class Parttime : public Employee

{

public :

int nowhr,hourly\_rate,salary;

void setdata()

{

cout<<"-------------------------"<<endl;

cout<<"Employee ID : ";

cin>>empcode;

cout<<"Employee Name : ";

fflush(stdin);

fgets(name,20,stdin);

name[strlen(name)-1]=0;

cout<<"Employee Hourly Rate : ";

cin>>hourly\_rate;

cout<<"No of Hours : ";

cin>>nowhr;

cout<<"\nSalary : "<<hourly\_rate\*nowhr<<endl;

cout<<"-------------------------"<<endl;

}

int getempid()

{

return empcode;

}

void show()

{

cout<<"-------------------------"<<endl;

cout<<"Status : Part Time"<<endl;

cout<<"Employee ID : "<<empcode<<endl;

cout<<"Employee Name : "<<name<<endl;

cout<<"Employee Hourly Rate : "<<hourly\_rate<<endl;

cout<<"No of Hours : "<<nowhr<<endl;

cout<<"\nSalary : "<<hourly\_rate\*nowhr<<endl;

cout<<"-------------------------"<<endl;

}

};

int main()

{

Fulltime F[10];

Parttime P[10];

int n,n1,a=0,b=0,empid,lim,i;

do

{

cout<<"1. Enter Record"<<endl;

cout<<"2. Display Record"<<endl;

cout<<"3. Search Record"<<endl;

cout<<"4. Quit"<<endl;

cout<<"\n Enter Your Choice : ";

cin>>n;

switch(n)

{

case 1 :

cout<<"1. Full time"<<endl<<"2. Part time"<<endl;

cin>>n1;

switch(n1)

{

case 1 :

F[a].setdata();

a++;

break;

case 2 :

P[b].setdata();

b++;

break;

default :

cout<<"Invalid Choice !"<<endl;

}

break;

case 2 :

if(a==0&&b==0)

cout<<"No Records!!"<<endl;

for(int i=0;i<a;i++)

{

F[i].show();

}

for(int i=0;i<b;i++)

{

P[i].show();

}

break;

case 3 :

cout<<"Enter Employee ID : ";

cin>>empid;

if(empid==0)

{

cout<<"No Employee found"<<endl;

}

else

{

lim=a>b?a:b;

for(i=0;i<lim;i++)

{

if(F[i].getempid()==empid)

{

F[i].show();

break;

}

if(P[i].getempid()==empid)

{

P[i].show();

break;

}

}

if(i==lim)

cout<<"No Employee found"<<endl;

}

break;

case 4 :

return 0;

default :

cout<<"Invalid Choice !";

cout<<"Try again : ";

}

}while(n!=4);

return 0;

}

8 - In a bank, different customers have savings account. Some customers may

have taken a loan from the bank. So bank always maintain information about

bank depositors and borrowers.

Design a Base class Customer (name, phone-number). Derive a class

Depositor(accno, balance) from Customer.

Again, derive a class Borrower (loan-no, loan-amt) from Depositor.

Write necessary member functions to read and display the details of ‘n’

customers.

Sol – 8.

#include<iostream>

#include<cstring>

using namespace std;

class Customer

{

private :

char name[20];

long long ph;

public :

void acceptc()

{

cout<<"-----------------------"<<endl;

cout<<"Enter Customer Name : ";

fflush(stdin);

fgets(name,20,stdin);

name[strlen(name)-1]=0;

cout<<"Enter Phone No : ";

cin>>ph;

}

void displayc()

{

cout<<"-----------------------"<<endl;

cout<<"Customer Name : "<<name<<endl;

cout<<"Phone No : "<<ph<<endl;

}

};

class Depositor : public Customer

{

private :

int acc,bal;

public :

void acceptd()

{

cout<<"Enter Account No : ";

cin>>acc;

cout<<"Enter Balance : ";

cin>>bal;

}

void displayd()

{

cout<<"Account No : "<<acc<<endl;

cout<<"Balance : "<<bal<<endl;

}

};

class Borrower : public Depositor

{

private :

int lno,lamt;

public :

void acceptb()

{

cout<<"Enter Loan No : ";

cin>>lno;

cout<<"Enter Loan Amount : ";

cin>>lamt;

}

void displayb()

{

cout<<"-----------------------"<<endl;

cout<<"Loan No : "<<lno<<endl;

cout<<"Loan Amount : "<<lamt<<endl;

cout<<"-----------------------"<<endl;

}

};

int main()

{

int n,i;

Borrower \*B;

cout<<"Enter the no of Customer : ";

cin>>n;

B=new Borrower[n];

for(i=0;i<n;i++)

{

B[i].acceptc();

B[i].acceptd();

B[i].acceptb();

}

cout<<"\n\n\*CUSTOMER DETAILS\*\n\n";

for(i=0;i<n;i++)

{

B[i].displayc();

B[i].displayd();

B[i].displayb();

}

return 0;

}

9. Write a C++ program to implement the following class hierarchy:

Student: id, name

StudentExam (derived from Student): Marks of 6 subjects

StudentResult (derived from StudentExam) : percentage

Define appropriate functions to accept and display details.

Create 'n' objects of the StudentResult class and display the marklist.

Sol – 9.

#include<iostream>

#include<cstring>

using namespace std;

class Student

{

private :

char name[20];

int id;

public :

void acceptc()

{

cout<<"-----------------------"<<endl;

cout<<"Enter Student Name : ";

fflush(stdin);

fgets(name,20,stdin);

name[strlen(name)-1]=0;

cout<<"Enter ID : ";

cin>>id;

}

protected :

void displayc()

{

cout<<"-----------------------"<<endl;

cout<<"ID : "<<id<<endl;

cout<<"Student Name : "<<name<<endl;

}

};

class StudentExam : public Student

{

private :

int marks[6];

public :

void acceptd()

{

for(int i=0;i<6;i++)

{

cout<<"Enter Marks of Subject "<<i+1<<" : ";

cin>>marks[i];

cout<<endl;

}

}

protected :

void displaym()

{

cout<<"\n\n";

for(int i=0;i<6;i++)

{

cout<<"Marks of Subject "<<i+1<<" : "<<marks[i]<<endl;

}

}

int getmarks(int n)

{

return marks[n];

}

};

class StudentResult : public StudentExam

{

private :

int sum=0;

float percent;

public :

void display()

{

for(int i=0;i<6;i++)

{

sum=sum+getmarks(i);

}

percent=(sum/600.0)\*100;

displayc();

displaym();

cout<<"\nPercentage : "<<percent<<endl;

cout<<"-----------------------"<<endl;

}

};

int main()

{

int n;

StudentResult \*S;

cout<<"Enter how many Students : ";

cin>>n;

S=new StudentResult[n];

for(int i=0;i<n;i++)

{

S[i].acceptc();

S[i].acceptd();

}

cout<<"\n\n\*\*MARKSHEET\*\*\n\n";

for(int i=0;i<n;i++)

{

S[i].display();

}

return 0;

}

10. Consider two base classes

worker(int code, char name, float salary),

officer(float DA, HRA)

class manger(float TA(is 10% of salary), gross salary) is derived from both base

classes.

Write necessary member functions.

Sol – 10.

#include<iostream>

#include<cstring>

using namespace std;

class Worker

{

int code;

char name[20];

protected :

float salary;

public :

Worker(){}

Worker(int c,char\* n,float s)

{

code=c;

strcpy(name,n);

salary=s;

}

void showw()

{

cout<<"---------------------------"<<endl;

cout<<"Code : "<<code<<endl;

cout<<"Name : "<<name<<endl;

cout<<"Salary : "<<salary<<endl;

}

};

class Officer

{

protected :

float da,hra;

public :

Officer(){}

Officer(float d,float h)

{

da=d;

hra=h;

}

void showo()

{

cout<<"DA : "<<da<<endl;

cout<<"HRA : "<<hra<<endl;

}

};

class Manager : public Worker, public Officer

{

float ta=salary/10.0;

float gs=salary+da+hra+ta;

public :

Manager(){}

Manager(int c,char\* n,float s,float d,float h):Worker(c,n,s),Officer(d,h)

{

}

void showm()

{

showw();

showo();

cout<<"TA : "<<ta<<endl;

cout<<"\nGross Salary : "<<gs<<endl;

cout<<"---------------------------"<<endl;

}

};

int main()

{

int i,n,c;

float sal,d,h;

char s[20];

cout<<"How many workers : ";

cin>>n;

Manager \*M;

M=new Manager[n];

for(i=0;i<n;i++)

{

cout<<"---------------------------"<<endl;

cout<<"Enter Code : ";

cin>>c;

cout<<"Enter Name : ";

fflush(stdin);

fgets(s,20,stdin);

s[strlen(s)-1]=0;

cout<<"Enter Salary : ";

cin>>sal;

cout<<"Enter DA : ";

cin>>d;

cout<<"Enter HRA : ";

cin>>h;

M[i]=Manager(c,s,sal,d,h);

}

cout<<"---------------------------"<<endl;

cout<<"\*\*\*WORKER INFO\*\*\*"<<endl;

cout<<"---------------------------"<<endl;

for(i=0;i<n;i++)

{

M[i].showm();

}

return 0;

}