



KARPAGAM ACADEMY OF HIGHER EDUCATION
 (Deemed University Established Under Section 3 of UGC Act 1956)
 Coimbatore - 641021.
 (For the candidates admitted from 2017 onwards)
DEPARTMENT OF COMMERCE (CA)

SUBJECT : PRACTICAL-OBJECT ORIENTED PROGRAMMING WITH C++

SEMESTER : I

SUBJECT CODE: 17CCU111

CLASS : I M.COM CA

MODEL QUESTION PAPER

1. Pay Roll calculation (Using simple program)
 2. Calculate Simple Interest and compound interest using inline functions
-
1. Find out EOQ, Minimum Level, Maximum Level, Re-order level (Using simple program)
 2. Write a c++ program to calculate the working capital using class and objects (member Function should be write inside and outside the class)
-
1. Calculate Depreciation – by using constructors and Destructors
 2. Calculation of BEP and MOS (give your own data).
-
1. Perform bank transactions like deposit, withdrawal, interest's calculation – by using classes and objects.
 2. Write a program to calculate funds from operation using (member function outside the class)
-
1. Write a perform to calculate overall cost of capital
 2. Write a c++ program to prepare the cost sheet using inheritance
-
1. Write a C++ program to calculate the sum and product numbers using operator overloading.
 2. Write a perform to calculate overall cost of capital
-
1. Program to create the student files and prepare the marks slips by accessing the File.
 2. Calculate Simple Interest and compound interest using inline functions
-
1. Write a perform to calculate overall cost of capital
 2. Calculation of BEP and MOS (give your own data).

1. Calculate Simple Interest and compound interest using inline functions
2. Write a perform to calculate overall cost of capital

1. Pay Roll calculation (Using simple program)
2. Write a program to calculate funds from operation using (member function outside the class)

1. Write a perform to calculate overall cost of capital
2. Calculation of BEP and MOS (give your own data).

1. EMPLOYEE PAYROLL CALCULATION

AIM:-

To find payroll calculation using simple program.

ALGORITHM:-

Step 1: Start the process

Step 2: Declare the variable as,bp,da,hra,pf,gpay,npay,esi.

Step 3: Using the “ICON” statement get the values for the variables.

Step 4: Calculate gross pay:

$$\text{Gpay} = \text{bp} + \text{da} + \text{hra}.$$

Step 5: Calculate net pay :

$$\text{Npay} = \text{gpay} - (\text{esi} + \text{pf})$$

Step 6: Using “cout” statement print the values in output screen.

Step 7: Save and run the program.

Step 8: Stop the process.

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
class employee
{
int empno;
char empname[20];
float bp,da,hra,pf;
float npay,gpay,esi;
public:
void getdetails();
void findnetsal();
void showempdetails();
};
void employee::getdetails()
{
cout<<"\n enter the employee number:";
cin>>empno;
cout<<"\n enter the employee name:";
cin>>empname;
cout<<"\n enter the basic pay:";
cin>>bp;
}
void employee::findnetsal()
{
```

```

da=0.15*bp;
hra=0.20*bp;
pf=0.15*bp;
esi=1.75*bp;
gpay=bp+da+hra;
npay=gpay-(esi+pf);
}
void employee::showempdetails()
{
cout<<"employee payroll details";
cout<<"\n enter the employee number:"<<empno;
cout<<"\n enter the employee name:"<<empname;
cout<<"\n enter the basic pay:"<<bp;
cout<<"\n enter the dearness allowance:"<<da;
cout<<"\n enter the house rent allowance:"<<hra;
cout<<"\n enter the provident fund:"<<pf;
cout<<"\n enter the esi:"<<esi;
cout<<"\n enter the grosspay:"<<gpay;
cout<<"\n enter the netpay:"<<npay;
}
void main()
{
clrscr();
employee emp[10];
int num,i;
cout<<"enter the number:";
cin>>num;
for(i=0;i<num;i++)
emp[i].getdetails();
for(i=0;i<num;i++)
emp[i].findnetsal();
for(i=0;i<num;i++)
emp[i].showempdetails();
getch();
}

```

OUTPUT

```
enter the number:1
enter the employee number:12
enter the employee name:keerthana
enter the basic pay:10000
employee payroll details
enter the employee number:12
enter the employee name:keerthana
enter the basic pay:10000
enter the dearness allowance:1500
enter the house rent allowance:2000
enter the provident fund:1500
enter the esi:17500
enter the grosspay:13500
enter the netpay:-5500_
```

RESULT: Hence the above program is verified.

2. ECONOMIC ORDER QUANTITY

AIM:-

To find economic order quantity, minimum level, maximum level, reorder level using simple program.

ALGORITHM:-

Step 1: Start the process

Step 2: Declared the variable as ml,rl,eqq,mml,d,s,h,i,c,nc,mmc,mc,rq using float data type and n* p ,m*p,mmp using int data type.

Step 3: Calculate Economic ordering quantity, maximum level, minimum level, reorder level, using class(member function is called outside the class).

Step 4: Calculate Economic order quantity

$$EOQ = \sqrt{2 * P * S / 4}$$

Step 5: Calculate maximum level=reorder level+ reorder quantity-(minimum consumption *minimum reorder period)

Step 6: Calculate maximum level=reorder level-(normal consumption*normal reorder period)

Step 7: Save the program

Step 8: Stop the process.

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
class eqq
{
float ml,rl,eqq,mml;
float d,s,h,i,c,nc,mmc,mc,rq;
int nrp,mrp,mmp;
public:
void getdata();
void finddata();
void showdata();
};
void eqq::getdata()
{
cout<<"\nenter the value of annual demand:";
cin>>d;
cout<<"\nenter the value of cost per order:";
```

```

cin>>s;
cout<<"\nenter the value of holding cost in percentage:";
cin>>i;
cout<<"\nenter the value of cost:";
cin>>c;
cout<<"\nenter the value of reorder quantity:";
cin>>rq;
cout<<"\nenter the value of minimum consumption:";
cin>>mmc;
cout<<"\nenter the value of minimum reorder period:";
cin>>mmp;
cout<<"\nenter the value of normal consumption:";
cin>>nc;
cout<<"\nenter the value of normal reorder period:";
cin>>npr;
cout<<"\nenter the value of maximum consumption:";
cin>>mc;
cout<<"\nenter the value of maximum reorder period:";
cin>>mrp;
}
void eoq::finddata()
{
    getdata();
    h=i*c;
    eoq=sqrt((2*d*s)/h);
    rl=mc*mrp;
    ml=rl+rq-(mmc*mmp);
    mml=rl-(nc*npr);
}
void eoq::showdata()
{
    cout<<"\neconomic ordering quantity"<<eoq;
    cout<<"\nmaximum level"<<ml;
    cout<<"\nminimum level"<<mml;
    cout<<"\nreorder level"<<rl;
}
void main()
{
    eoq e;
    clrscr();
    e.finddata();
    e.showdata();
    getch();
}

```

OUTPUT

```
enter the value of cost per order:500
enter the value of holding cost in percentage:10
enter the value of cost:5
enter the value of reorder quantity:600
enter the value of minimum consumption:500
enter the value of minimum reorder period:5
enter the value of normal consumption:500
enter the value of normal reorder period:5
enter the value of maximum consumption:600
enter the value of maximum reorder period:6

economic ordering quantity447.213593
maximum level1700
minimum level1100
reorder level3600_
```

RESULT: Hence the above program is verified.

3. WORKING CAPITAL

AIM:-

Write a C++ program to calculate the working capital using classes & objects (Member should be like write inside and outside the class)

ALGORITHM:-

Step 1: Start the process

Step 2: Declare the variables as a1,l1 ,a2, l2, wc1, wc2.

Step 3: Using the “Cin” statement get the value for the variable.

Step 4: Calculate Working capital

$$wc1=a1-a2$$

$$wc2=a2-l2.$$

Step 5: Using the “COUT” statement print values in output screen.

Step 6: Save and run the program.

Step 7: Stop the program.

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
class workcap
{
long int a1,a2,l1,l2,wc1,wc2;
public:
void getdata()
{
cout<<"\n\t\t\tworking capital";
cout<<"\nenter the previous year current assets :rs.";
cin>>a1;
cout<<"\nenter the previous year current liability:rs.";
cin>>l1;
cout<<"\nenter the current year current assets:rs.";
cin>>a2;
cout<<"\nenter the current year current liability:rs.";
cin>>l2;
}
void calculate(void);
};
void workcap::calculate()
{
long int temp;
wc1=a1-l1;
```

```

wc2=a2-l2;
cout<<"\nprevious year working capital:rs."<<wc1;
cout<<"\ncurrent year working capital:rs."<<wc2;
if(wc1>wc2)
{
temp=wc1-wc2;
cout<<"\n\nDecreasing working capital:rs."<<temp;
}
else
{
temp=wc1-wc2;
cout<<"\n\nIncreasing working capital:rs."<<temp;
}
}
void main()
{
clrscr();
workcap x;
x.getdata();
x.calculate();
getch();
}

```

OUTPUT

```

              working capital
enter the previous year current assets :rs.100000

enter the previous year current liability:rs.25000

enter the current year current assets:rs.200000

enter the current year current liability:rs.150000

previous year working capital:rs.75000
current year working capital:rs.50000

Decreasing working capital:rs.25000_

```

RESULT: Hence the above program is verified.

4. SIMPLE AND COMPOUND INTEREST

AIM:-

To calculate simple interest & compound interest using inline function.

ALGORITHM:-

Step 1: Start the process

Step 2: Declare the variables as p, n, r, si, ci.

Step 3: Using the “CIN” statement get the value for the variables.

Step 4: Calculate simple and compound interest

$$si = p * n * r / 100$$

$$ci = p * \text{pow}(1 + r / 100, n)$$

Step 5: Using the “COUT” statement print the value in output.

Step 6: Save and run the program

Step 7: Stop the program.

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
class interest
{
int p,n,r;
float si,ci;
public:
void getdata();
void finddata();
void showdata();
};
inline void interest ::getdata()
{
cout<<"\n enter the principle value:";
cin>>p;
cout<<"\n enter the no of years:";
cin>>n;
cout<<"\n enter the rate of interest:";
cin>>r;
}
inline void interest::finddata()
{
si=p*n*r/100;
ci=p*pow(1+r/100,n);
```

```

}
inline void interest::showdata()
{
cout<<"\n SI & CI:";
cout<<"\n *****";
cout<<"\n simple interest:"<<si;
cout<<"\n compound interest:"<<ci;
}
void main()
{
clrscr();
interest x;
x.getdata();
x.finddata();
x.showdata();
getch();
}

```

OUTPUT

```

enter the principle value:1000
enter the no of years:5
enter the rate of interest:5
SI & CI:
*****
simple interest:250
compound interest:1000_

```

RESULT: Hence the above program is verified.

5. CONSTRUCTOR AND DESTRUCTOR

AIM:-

To calculate depreciation by constructor and destructor.

ALGORITHM:-

Step 1: Start the program

Step 2: Declare the variable as c ,s, r, y, amt, rate.

Step 3: Get the values for the variables using “CIN” statement.

Step 4: Calculate depreciation using

(i)Diminishing balance method

$amt = (1-s)/y;$

$rate = amt/c*100;$

$amt = c*r/100;$

$c = c - amt;$

(i)Straight line method

$c = c - amt.$

Step 5: Using the “COUT” statement print the value in output screen

Step 6: Save the program

Step 7: Stop the program

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
class dep
{
private:
float c,s,r,y,amt,amount,rate;
public:
void calculate();
void showresult();
void depreciation();
~dep();
};
void dep::depreciation()
{
cout<<"\n\t depreciation:";
cout<<"\n enter the cost of asset:";
cin>>c;
cout<<"\n enter the scrape value:";
```

```

cin>>s;
cout<<"\n enter the estimated life(only in years):";
cin>>y;
cout<<"\n enter the rate of depreciation:";
cin>>r;
}
void dep::~dep()
{
}

void dep::calculate()
{
amount=(c-s)/y;
rate=amount/c*100;
cout<<"\n\t diminishing balance method:";
cout<<"\n\t *****:";
for(int i=1;i<=y;i++)
{
amt=c*r/100;
cout<<"\n the amount of depreciation year:"<<c<<"amt"<<endl;
c=c-amt;
}
}
void dep::showresult()
{
cout<<"\n\t straight line method:";
cout<<"\n the amount of depreciation for all the year:"<<amount;
cout<<"\n rate of depreciation"<<rate;
}
void main()
{
clrscr();
dep r;
r.depreciation();
r.calculate();
r.showresult();
getch();
}

```

OUTPUT

```
depreciation:
enter the cost of asset:500000
enter the scrape value:20000
enter the estimated life(only in years):5
enter the rate of depreciation:10

diminishing balance method:
*****:
the amount of depreciation year:500000amt
the amount of depreciation year:450000amt
the amount of depreciation year:405000amt
the amount of depreciation year:364500amt
the amount of depreciation year:328050amt

straight line method:
the amount of depreciation for all the year:96000
rate of depreciation19.200001
```

RESULT: Hence the above program is verified.

6. OPERATOR OVERLOADING

AIM:-

Write a C++ program to calculate the sum and product number using operator overloading

ALGORITHM:-

Step 1: Start the program

Step 2: Declare the variables as a, b, c, ch, opt.

Step 3: Using the "CIN" statement the value for the variable

Step 4: Calculate addition, multiplication using operators overloading

(i) $m+m, m*m$.

(i) $m.c=m, a*m.b$.

Step 5: Using the "COUT" statement print the value in output screen

Step 6: Save and run the program.

Step 7: Stop the program.

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
class over
{
public:
int a,b,c;
void read();
void operator+(over);
void operator*(over);
};
void main()
{
over x;
int ch,opt;
clrscr();
cout<<"\n\t\t\t operator overloading:";
cout<<"\n\t\t\t ~~~~~~:";
x.read();
do
{
cout<<"\n1. addition:";
cout<<"\n2. multiplication:";
cout<<"\n\n select operationto perform:";
cin>>opt;
```

```

switch(opt)
{
case 1:x+x;
break;
case 2:x*x;
break;
default:cout<<"\n sorry invalid operator:";
}
cout<<"\n\t do you want to continue(1.yes/2.no):";
cin>>ch;
}
while(ch!=0);
}
void over::read()
{
cout<<"\n enter the integer value for a:";
cin>>a;
cout<<"\n enter the integer value for b:";
cin>>b;
}
void over::operator+(over x)
{
cout<<"\n\n addition:";
x.c=x.a+x.b;
cout<<x.c<<"\n";
}
void over::operator*(over x)
{
cout<<"\n\n multiplication:";
x.c=x.a*x.b;
cout<< x.c<<"\n";
}

```

OUTPUT

```
operator overloading:
~~~~~:

enter the integer value for a:20

enter the integer value for b:10

1. addition:
2. multiplication:

select operationto perform:1

addition:30

do you want to continue(1.yes/2.no):1

1. addition:
2. multiplication:

select operationto perform:2

multiplication:200

do you want to continue(1.yes/2.no):0
```

RESULT: Hence the above program is verified.

7. COST SHEET USING INHERITANCE

AIM:-

To write a C++ program for cost sheet using inheritance.

ALGORITHM:-

Step 1: Start the program

Step 2: Declare the variables as fc, pc, fo, oo, so, s, cp, cs, dm, de, dw, profit;

Step 3: Using "CIN" statement the value for the variables.

Step 4: calculate prime cost, cost of production, cost of sales, factory overhead using

pc=dm+dw+de;

fc=pc+fc;

cp+fc+100;

profit=s-cs;

Step 5: Using the "COUT" statement print the value in output screen.

Step 6: Save and run the program

Step 7: Stop the program.

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
class cost
{
protected:
long double fc,pc,fo,oo,so,s;
public:
void getdata()
{
cout<<"\n enter the factory overheads:";
cin>>fo;
cout<<"\n enter the office overheads:";
cin>>oo;
cout<<"\n enter the selling overheads:";
cin>>so;
cout<<"\n enter the sales:";
cin>>s;
}
void show()
{
cout<<"\n_____";
cout<<"\n\nprime cost:"<<pc;
```

```

cout<<"\n\n(+)factory overheads:"<<fo;
cout<<"\n\nfactory cost:"<<fc;
cout<<"\n\n(+)office overheads:"<<oo;
}
};
class sheet:public cost
{
protected:
long double cp,cs,dm,de,dw,profit;
public:
void get()
{
cout<<"\n\t\t\tcost sheet";
cout<<"\n\t\t\t*****";
cout<<"\n\n enter the direct material:";
cin>>dm;
cout<<"\n\nenter the direct wages:";
cin>>dw;
cout<<"\n\nenter the direct expenses:";
cin>>de;
}
void calculate()
{
pc=dm+dw+de;
fc=pc+fo;
cp=fc+oo;
cs=cp+so;
profit=s-cs;
}
void display()
{
cout<<"\n\ncost of production:"<<cp;
cout<<"\n\n(+)selling overhead:"<<so;
cout<<"\n\ncost of sales:"<<cs;
cout<<"\n\n(-)sales:"<<s;
cout<<"\n\n-----";
if(s>cs)
{
cout<<"\n\nprofit="<<profit;
}
else
{
cout<<"\n\nloss="<<profit;
}
cout<<"\n\n-----:";
}
};
void main()
{
clrscr();

```

```
sheet s1;  
s1.get();  
s1.getdata();  
s1.calculate();  
s1.show();  
s1.display();  
getch();  
}
```

OUTPUT

```
enter the selling overheads:20000  
  
enter the sales:100000  
  
-----  
prime cost:50000  
  
(+)factory overheads:10000  
  
factory cost:60000  
  
(+)office overheads:5000  
  
cost of production:65000  
  
(+)selling overhead:20000  
  
cost of sales:85000  
(-)sales:100000  
-----  
profit=15000  
-----:_
```

RESULT: Hence the above program is verified.

8. BREAK EVEN POINT

AIM:

To Calculate Break Even Point & Margin of Safety.

ALGORITHM:

Step 1: Start the process

Step 2 : Declare the variable as sales, vc, fc, profit, cont, ratio, safety, dep.

Step 3: Using the 'cin' statement the values for the variable.

Step 4: Calculate BEP & MOS using

$$\text{PROFIT} = (\text{sales} - \text{vc} - \text{fc})$$

$$\text{CONT} = \text{sales} - \text{vc}$$

$$\text{RATIO} = \text{cont} / \text{sales} * 100$$

Step 5: Using the 'cout' statement print the value in the output screen

Step 6: Save and run the program

Step 7: Stop the program

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{
clrscr();
int tfc,c,as,bep,mos;
cout<<"\n enter the value for total fixed cost:";
cin>>tfc;
cout<<"\n enter the value of contribution/unit:";
cin>>c;
cout<<"\n enter the value of actual sales:";
cin>>as;
bep=tfc/c;
mos=as-bep;
cout<<"\n break even point:"<<bep;
cout<<"\n margin of safety:"<<mos;
getch();
}
```

OUTPUT

```
enter the value for total fixed cost:2000
enter the value of contribution/unit:50
enter the value of actual sales:30000
break even point:40
margin of safety:29960_
```

RESULT: Hence the above program is verified.

9. BANK TRANSACTION

AIM:

To perform bank transactions like depreciation withdraw and interest calculation by using classes and objects.

ALGORITHM:

Step 1: Start the process

Step 2: Create a variable as a/c no, ch, bal, withdraw.

Step 3: using the switch statement perform the function calculation.

Step 4: incase bal + dep is used and the operator and it calls the function operator

bal = bal +dep.

Step 5: incase to bal – withdrawal is used and the operator and it calls the function operator

Step 6: incase 3, point the current balance in account

Step 7: incase 4, exit output screen

Step 8: It has used to either 1 or 2, using the ch, variable otherwise the default statement will
be executed

Step 9: The do while loop is used for interest calculation

Step 10: Stop the process

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
class bank
{
char name[20],acc[10];
int accno,ch;
float bal,dep,withdrawl;
public:
bank();
};
void bank::bank()
{
cout<<"\n enter the name of the customer:";
cin>>name;
```

```

cout<<"\n enter ther acc no:";
cin>>accno;
cout<<"\n balance amount:rs";
cin>>bal;
do
{
cout<<"\n1.deposite:";
cout<<"\n2.withdrawel:";
cout<<"\n3.current balance:";
cout<<"\n4.exit:";
cout<<"\n enter your choice:";
cin>>ch;
switch(ch)
{
case 1:
cout<<"\n enter ther amount to be deposited:rs";
cin>>dep;
bal=bal+dep;
cout<<"\n enter the current balance:rs"<<bal;
break;
case 2:
cout<<"\n enter the amount to be withdrawel:rs";
cin>>withdrawl;
bal=bal-dep;
cout<<"\n enter the current balance is:rs"<<bal;
break;
case 3:
cout<<"\n after transaction the current balance is"<<bal;
break;
case 4:
cout<<"\n exit:";
break;
default:
cout<<"\n invalid choice:";
}
}
while(ch!=4) ;
}
void main()
{
clrscr();
bank x;
getch();
}

```

OUTPUT

```
enter the current balance:rs90000
1.deposite:
2.withdrawel:
3.current balance:
4.exit:
enter your choice:2

enter the amount to be withdrawel:rs20000

enter the current balance is:rs60000
1.deposite:
2.withdrawel:
3.current balance:
4.exit:
enter your choice:3

after transaction the current balance is60000
1.deposite:
2.withdrawel:
3.current balance:
4.exit:
enter your choice:4

exit:_
```

RESULT: Hence the above program is verified.

10. STUDENT INFORMATION SYSTEM USING FILES

AIM:

Write a program to create the student files and mark slip by files operations

ALGORITHM:

Step 1: Start the program

Step 2 : Declare the variable name [30],roll no,avg,m1,m2,m3,m4,m5,total,m,avg,result[10];

Step 3: Open a file name student in the output mode

Step 4: Calculate total & avg using

$$\text{total} = m1 + m2 + m3 + m4 + m5$$
$$\text{avg} = \text{total} / 5$$

Step 5: Using the 'cout' statement print the value in the output screen

Step 6: Save and run the program

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<fstream.h>
void main()
{
    ofstream out("student");
    char name[25];
    int regno;
    long int m1,m2,m3,m4,m5,total,m;
    float avg;
    char result[10];
    clrscr();
    cout<<"\n\t student file:";
    cout<<"\n enter the name of the student:";
    cin>>name;
    cout<<"\n enter the regno:";
    cin>>regno;
    cout<<"\n enter the tamil mark:";
    cin>>m1;
```

```

cout<<"\n enter the cf mark:";
cin>>m2;
cout<<"\n enter the me mark:";
cin>>m3;
cout<<"\n enter the OR mark:";
cin>>m4;
cout<<"\n enter ther english mark:";
cin>>m5;
total=(m1+m2+m3+m4+m5);
avg=(total/5);
if((m1>40)&&(m2>40)&&(m3>40)&&(m4>40)&&(m5>40))
{
strcpy(result,"pass");
}
else
{
strcpy(result,"fail");
}
cout<<name<<"\n"<<regno<<"\n"<<m1<<"\n"<<m2<<"\n"<<m3<<"\n"<<m4<<"\n"<<m5<<"\n";
ifstream in("student");
clrscr();
cout<<"\n\t mark statement:";
cout<<"\n\t *****:";
if(in.eof()==0)
{
cout<<"\n name:"<<name;
cout<<"\n regno:"<<regno;
cout<<"\n mark1:"<<m1;
cout<<"\n mark2:"<<m2;
cout<<"\n mark3:"<<m3;
cout<<"\n mark4:"<<m4;
cout<<"\n mark5:"<<m5;
cout<<"\n total:"<<total;
cout<<"\n average:"<<avg;
cout<<"\n _____:";
cout<<"\n result:"<<result;
cout<<"\n #####:";
}
in.close();
getch();
}

```

OUTPUT

```
      student file:
enter the name of the student:revathi

enter the regno:003

enter the tamil mark:90

enter the cf mark:95

enter the me mark:78

enter the OR mark:85

enter ther english mark:75
```

```
      mark statement:
**** *****:
name:revathi
regno:3
mark1:90
mark2:95
mark3:78
mark4:85
mark5:75
total:423
average:84
_____:
result:pass
#####:
```

RESULT: Hence the above program is verified.

11. OVERALL COST OF CAPITAL

AIM:

To write c++ program to perform overall cost of capital.

ALGORITHM:

Step 1: Start the program

Step2: Declare the variable as ebit,v,ko

Step 3: Using the cin statement value for the variable

Step 4: Calculate overall cost of capital $ko = ebit/v$

Step 5: Using the cout statement print the value in the output screen

Step 6: Save and run the program.

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{
double ebit,v,ko;
clrscr();
cout<<"\n enter the value of earnings before interest and tax:";
cin>>ebit;
cout<<"\n enter the value of the firm:";
cin>>v;
ko=ebit/v;
cout<<"\n overall cost of capital:"<<ko;
getch();
}
```

OUTPUT

```
enter the value of earnings before interest and tax:2000  
enter the value of the firm:5  
overall cost of capital:400
```

RESULT: Hence the above program is verified.

12. FUND FLOW OPERATION

AIM:

Write a program to calculate funds flow operation using member function outside the class.

ALGORITHM:

Step 1: Start the program

Step 2: Declare the variable as calc, ni, ii, ie, dep, gain, loss

Step 3: Using the cin statement get the values for variables

Step 4: Calculate fund flow operation = net income-interest income+ interest expenses+depreciation-gain+loss.

Step 5: Using the cout statement print the value in the output screen

Step 6: Save and run the program

Step 7: Stop the program

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
class fund
{
float calc,ni,ii,ie,dep,gain,loss;
public:
void getdetails();
void findfunds();
void showfunddetails();
};
void fund::getdetails()
{
cout<<"\n enter the net income:";
cin>>ni;
cout<<"\n enter the interest income:";
cin>>ii;
cout<<"\n enter the interest expense:";
cin>>ie;
cout<<"\n enter the depreciation:";
cin>>dep;
cout<<"\n enter the gain on asset sales:";
cin>>gain;
cout<<"\n enter the loss on asset sales:";
cin>>loss;
```

```

}
void fund::findfunds()
{
getdetails();
calc=ni-ii+ie+dep-gain+loss;
}
void fund::showfunddetails()
{
cout<<" fund flow operation is:";
cout<<calc;
}
void main()
{
clrscr();
fund x;
x.findfunds();
x.showfunddetails();
getch();
}

```

OUTPUT

```

enter the net income:3000

enter the interest income:4000

enter the interest expense:2000

enter the depreciation:4000

enter the gain on asset sales:5000

enter the loss on asset sales:6000
fund flow operation is:6000_

```

RESULT: Hence the above program is verified.



KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed University Established Under Section 3 of UGC Act 1956)
Coimbatore - 641021.
(For the candidates admitted from 2017 onwards)
DEPARTMENT OF COMMERCE (CA)

		Semester – I			
		L	T	P	C
17CCP111	PRACTICAL -1 - C++	-	-	4	2

Program Outcome

- The course includes Class, Objects, Inline function, Constructors, Destructors, Operator Overloading and Inheritance

Program Learning Course

- To Know about Object Oriented concept
- To acquire knowledge in Functions of C++
- To develop practical skills by using C++ Program

1. Pay Roll calculation (Using simple program)
2. Find out EOQ, Minimum Level, Maximum Level, Re-order level (Using simple program)
3. Write a c++ program to calculate the working capital using class and objects (member Function should be write inside and outside the class)
4. Calculate Simple Interest and compound interest using inline functions.
5. Calculate Depreciation – by using constructors and Destructors
6. Write a C++ program to calculate the sum and product numbers using operator overloading.
7. Write a c++ program to prepare the cost sheet using inheritance
8. Calculation of BEP and MOS (give your own data).

9. Perform bank transactions like deposit, withdrawal, interest's calculation – by using classes and objects.
10. Program to create the student files and prepare the marks slips by accessing the File.
11. Write a perform to calculate over all cost of capital
12. Write a program to calculate funds from operation using (member function outside the class)