Branch - I.T. T2 Exam Oct./Nov. 2018. SY B.Tech.
Course Name - Problem Solving & object oriented programming
Course code - ITUA2176
Paper code - U218-146(12)

Marking Scheme

Q 1) a) Differentiate between object oriented and Procedural oriented language. [6]

At least 6 points

b) Create an employee class, the member data should comprise an int for storing the employee number and a float for storing the employee's compensation. Member functions should allow the user to enter this data and display it. Write a main() that allows the user to enter data for three employees and display it. [6]

Code with proper logic.

c) What are the rules for default argument? [4]

Explanation of rules with example

OR

Q2) a) Define constructor and destructor. Demonstrate dynamic constructer and destructor through example. [6]

Definition 2marks
Example 4 marks

b) Write a program in C++ to enter P, T, R and calculate Simple Interest.

Code with proper logic.

[6]

c) What is Constructor? How many types of constructors are there in C++ ? [4]

1 marks – definition 3marks – types of constructor

Q3) a) Write a program to swap values using friend function [6]

Code with proper logic.

b) Discuss the concept of virtual base class. [4]

Explanation with diagram

c) Which operators cannot be overloaded? [4]

1. Class member access operator (.)
2. Scope resolution operator(::)
3. Pointer to member operator (.*)
4. Conditional operator(?:)
5. Size of operator(sizeof)

OR

Q4) a) Write program to overload unary minus (-) operator using friend function. [6]

Code with proper logic.

b) List down type conversion supported by C++. How it is achieved?[4]

Type of type conversion – 2 marks, Way to achieve it-2marks

c) Explain need and use of inheritance in C++. [4]

- Reusability is another important feature of OOP
- Using the features(data and/or functions) of one class into another class
- Mechanism of deriving a new class from an old one is called as inheritance(or derivation)
- The old class is referred as Base(super) class and new class is called as derived(Sub) class

Branch - I.T. T2 Exam oct./Nov-2018 SYBTech.
Course Name - Problem Solving & Object Oriented Programming
Course Code - ITUA 2176
Paper Code - U218-146 (T2)

Solution

Q 1) a) Differentiate between object oriented and Procedural oriented language. [6]

Feature	Procedure oriented Programming	Object oriented Programming
Divided Into	In POP Program is divided into small parts called **functions**	In OOP, program is divided into parts called **objects**.
Importance	In POP, Importance is not given to **data** but to functions as well as **sequence** of actions to be done.	In OOP, Importance is given to the data rather than procedures or functions because it works as a **real world**.
Approach	POP follows **Top Down approach**.	OOP follows **Bottom Up approach**
Access Specifiers	POP does not have any access specifier	OOP has access specifiers named Public, Private, Protected, etc.
Data Moving	In POP, Data can move freely from function to function in the system.	In OOP, objects can move and communicate with each other through member functions.
Expansion	To add new data and function in POP is not so easy.	OOP provides an easy way to add new data and function
Data Access	In POP, Most function uses Global data for sharing that can be accessed freely from function to function in the system.	In OOP, data can not move easily from function to function, it can be kept public or private so we can control the access of data.
Data Hiding	POP does not have any proper way for hiding data so it is **less secure**.	OOP provides Data Hiding so provides **more security**.
Overloading	In POP, Overloading is not possible.	In OOP, overloading is possible in the form of Function Overloading and Operator Overloading.
Examples	Example of POP are : C, VB, FORTRAN, Pascal.	Example of OOP are : C++, JAVA, VB.NET, C#.NET.

b) Create an employee class, the member data should comprise an int for storing the employee number and a float for storing the employee's compensation. Member functions should allow the user to enter this data and display it. Write a main() that allows the user to enter data for three employees and display it. [6]

Code -

```
#include<iostream>
using namespace std;
class employee
{
        private:
        int emp_no;
        float compensation;
        public:
        void getdata();
        void putdata();
        employee ();
};
        void emploee::getdata()
        {
                cout<<"\nEmployee number =  ";
                cin>>emp_no;
                cout<<"Compensation  ";
                cin>>compensation;
        }
        void employee::putdata()
        {       cout<<"employee number"<<"\t";
                cout<<emp_no<<"\t";
                cout<<Compensation<<"\t";
                cout<<compensation;
        }
        employee::employee()
        {       emp_no = 0;
        compensation = 0.0
        }
        int main()
        {
                employee e[5];
                int a;
                do
                {
cout<<"Enter your choice  \n1.Getdata \n2.Putdata \n3.Exit \n Enter 1,2,3only \n";
                        cin>>a;

        cout<<"=====================================================\n";
                        switch(a)
                        {       case 1:
                                for(int i=0;i<5;i++)
                                {
                                        cout<<"Enter data for object \n";
```

```
                    e[i].getdata();
                }
                break;
case 2: cout<<"Employee Number\t"<<"Compensation\t"
cout<<"=================================================\n";
                for(int i=0;i<5;i++)
                {
                    e[i].putdata();
                }
                break;
                }
        }       while(a!=4);
}
```

c) What are the rules for default argument? [4]

- Once an argument has a default value, all the arguments after it must have default values.
- Once an argument is defaulted in a function call, all the remaining arguments must be defaulted.
- Example

int f(int x, **int** y=0, **int** n)	**int f(int** x, **int** y=0, **int** n=1)
// legal	// illegal

OR

Q2) Define constructor and destructor. Demonstrate dynamic constructer and destructor through example. [6]

- Constructor: - It is a special member function which initializes the objects of its class.
- Destructor:- It is a member function which deletes an object.

Example :-

```
class string {
 private:
   char *s;
   int size;
 public:
   string();
   string(char *);    // constructor
   ~string();         // destructor

};
string::string()
{
size=0;
```

```
s=new char[size+1];
}

string::string(char *c)
{
 size = strlen(c);
 s = new char[size+1];
 strcpy(s,c);
}
string::~string()
{
 delete []s;
}
```

b) Write a program in C++ to enter P, T, R and calculate Simple Interest. [6]

Code -

```
#include<iostream>
using namespace std;
class interest
{
        private:
        int P, R, T;
        public:
        void getdata();
        void calculate(interest);
        interest ();
};
        void interest::getdata()
        {
                cout<<"\n Principal Amount= ";
                cin>>P;
                cout<<"\n Rate of Interest ";
                cin>>R;
                cout<<"\n Time Period ";
                cin>>T;

        }
        void employee::calculate(interest I)
                float cal;
                cal = (I.P*I.R* I.T)/100
        {       cout<<"Calculated Interest is ="
                cout<<cal ;

        }
        interest::interest()
        {       P = 0;
                R=0;
                T=0;
        }
        int main()
        {
                Interest i1;
                int a;
```

```
        do
        {
cout<<"Enter your choice  \n1.Getdata \n2.Calculate Interest \n3.Exit \n";
                cin>>a;

cout<<"=====================================================\n";
                switch(a)
                {       case 1:
                        I1.getdata();
                        break;
                        case 2:  cout<<"Calculated Interrest"
                cout<<"=========================================\n";
                        i1.calculate(i);
                        break;
                        }
        }       while(a!=4);
}
```

c) What is Constructor? How many types of constructors are there in C++ ? [4]

Definition: It is a special member function which initializes the objects of its class.

- The constructor is invoked whenever an object of its class is created.
- It is called constructor because it constructs the value of data members of the class.
- A constructor has:

    (i) the same name as the class itself

    (ii) no return type

Types of Constructor

- If constructor accepts no parameters, is called default constructor & default constructor for class A is A::A().If no such constructor is defined, then the compiler supplies its own constructor.
- You must supply the arguments to the constructor when a new instance is created is called parametrizee constructor.
- It is a member function which initializes an object using another object of the same class.

- A copy constructor has the following general function prototype:

  class_name (const class_name&);

Q3) a) Write a program to swap values using friend function [6]

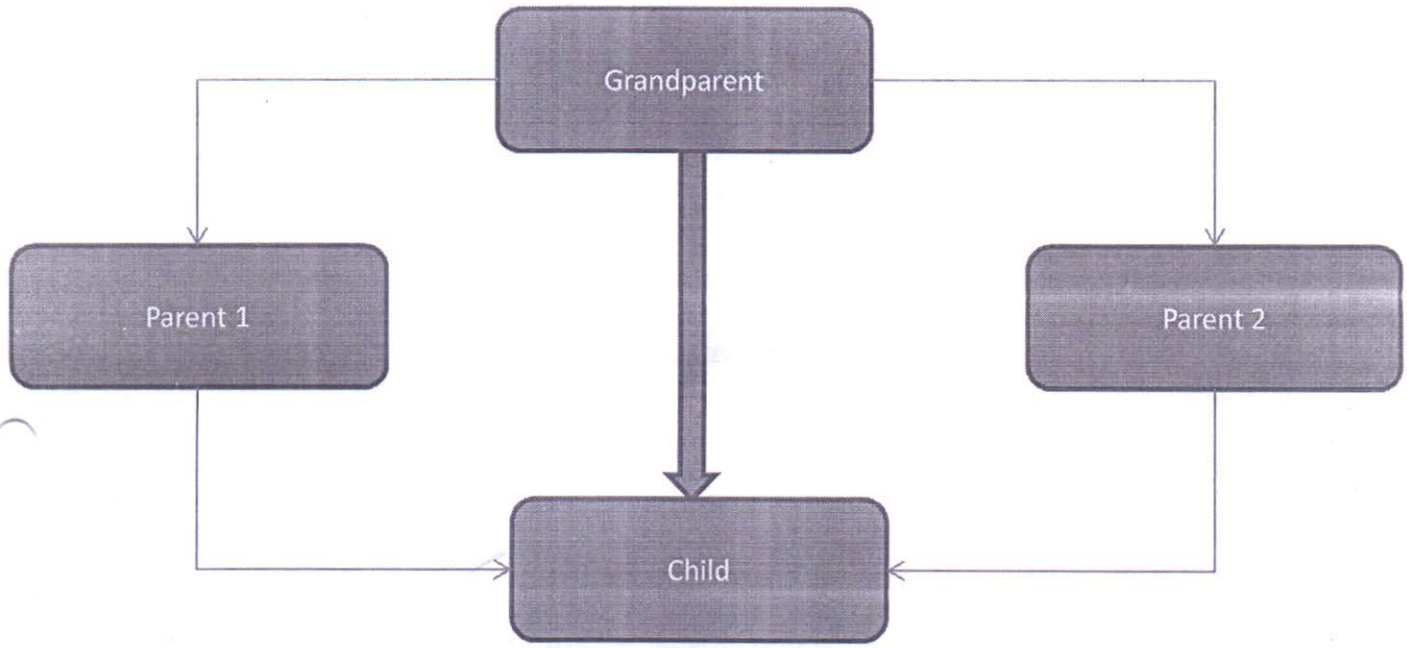
Code :-

```
class abc
{
    int a;
public:
    void set(int p)
    {
            a=p;
    }
    void show()
    {
            cout<<a<<endl;
    }
    friend void swap(xyz &,abc &);
};
void swap(xyz &xx,abc &aa)
{
    int temp;
    temp=xx.x;
    xx.x=aa.a ;
    aa.a =temp;
    cout<<xx.x <<aa.a <<endl;
}
void main()
{
    xyz x1;
    abc a1;
    a1.set (100);
    x1.set (500);
    cout<<"Before swap"<<endl;
    x1.show ();
    a1.show ();
    cout<<"After swap"<<endl;
    swap(x1,a1);
    cout<<"From main"<<endl;
    x1.show ();
    a1.show ();
}
```

b) Discuss the concept of virtual base class. [4]

- In above case all the public as well as protected members of grandparent are inherited 'twice', so child would have duplicate copies.

- Can be avoided by making common base class(grandparent in this case) as virtual base class



c) Which operators cannot be overloaded? [4]

1. Class member access operator (.)
2. Scope resolution operator(::)
3. Pointer to member operator (.*)
4. Conditional operator(?:)
5. Size of operator(sizeof)

OR

Q4) a) Write program to overload unary minus (-) operator using friend function. [6]

Code :-

```
Class data
{
        int x,y;
        public :
        void get(int a, int b);
        void display();
        friend void operator-(data );
};
void data :: get(int a, int b)
{
        x = a;
        y = b;
}
void data :: display()
{
```

```
        cout<<x<<" " ;
        cout<<y<<" " ;
}
void operator-(data d1){
        d1.x = -d1.x;
        d1.y = -d1.y;
}
int main()
{
        data d;
        d.get(10, -20);
        cout<< "d : ";
        d.display();

        -d; // activate operator funcn
        cout<<" d : ";
        d.display();
        return 0;
}
```

b) List down type conversion supported by C++. How it is achieved?[4]

- Three type of situations might arise in the data conversion between incompatible types :
  - Conversion from basic type to class type.
  - Conversion from class type to basic type
  - Conversion from one class type to another class type

Basic to class type

- The conversion from basic to class type is accomplish through constructor

Class type to basic

- The conversion from class to basic type is accomplish through overloaded casting operator
- Syntax

```
operator typename ()
{
        .....
```

..... (function statements)

}

- This function converts a class type data to typename.

One class to another class type

The conversion from class to other class is carried out by either constructor or conversion function

c) Explain need and use of inheritance in C++.

- Reusability is another important feature of OOP
- Using the features(data and/or functions) of one class into another class
- Mechanism of deriving a new class from an old one is called as inheritance(or derivation)
- The old class is referred as Base(super) class and new class is called as derived(Sub) class