Basic & Advance C++ Programs

Basics Concepts of C++, Tokens Expression and Control structures

1. Write a C++ program that will output this passage by Deepak Chopra. Make sure your output looks exactly as shown here (including spacing, line breaks, punctuation, and the title and author). **Use cout and cin objects and endl manipulator.**

```
******************************
*    Programmimg Assignment 1    *
*    Computer Programmimg I      *
*          Author : ???          *
*    Due Date: Thursday, Dec. 20 *
******************************
```

Question: Difference between \n and endl.

2. Write a program to create the following table. **Use endl and setw manipulator.**

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

3. Write a C++ program to add two floating numbers using pointer. The result should contain only two digits after the decimal.**Use fixed, scientific and setprecision () manipulators for controlling the precision of floating point numbers.**

4. Write a C++ program to find out sum of array element using Recursion.
Question: Show stepwise solution of **winding and unwinding phase of recursion**

5. Write a C++ program to find the number of vowels present in the given character array using pointer arithmetic.

6. Find error in the following code and give reasons for each error:
Can we declare an array of references? Can we assign NULL value to reference variable? Is **Reference variable** a pointer variable? Can we declare a reference variable without initializing it? Does Reference Variable change the original value of variable?

```
#include<iostream>
using namespace std;
int main()
{
   int no1=10, no2=12;
   int & x=no1;
   int & r;
   int & c = NULL;
   int & d[2] = {no1,no2};
   cout<<"x = "<< x+20;
   cout<<"no1="<< no1+10;
   return 0;
}
```

7. Find output of the following code: Explain how **scope Resolution operator** is used to access global version of a variable.

```
#include<iostream.h>
#include<conio.h>
int m=30;
int main()
{
   int m=20;
  {
     int m=10;
     cout<<"we are in inner block"<<endl;
     cout<<"value of m="<<m<<"\n";
     cout<<"value of ::m="<<::m<<"\n";
  }
  cout<<"we are in outer block"<<endl;
  cout<<"value of m="<<m<<"\n";
  cout<<"value of ::m="<<::m<<"\n";
  getch();
  return 0;}
```

8. Find Error in the following code of a program and give explanation why these errors exist.
Refer this link for more understanding:
http://www.thegeekstuff.com/2012/06/c-constant-pointers/
Note: There is mistake in Balagury 6th edition for syntax of constant pointer and pointer to constant.

1. //This is an example of **constant pointer**	**2.** //This is an example of **pointer to constant**	**3.** //This is an example of **constant pointer to a constant**

#include <iostream>
using namespace std;
int main()
{
int var1 = 35,var2 = 20;
int *const ptr = &var1;
ptr = &var2;
cout<<"var1= "<<*ptr;
return 0;
}
	#include <iostream>
	using namespace std;
	int main()
	{
	int var1 = 43;
	const int* ptr = &var1;
	*ptr = 1;
	var1=34;
	cout<<"var1 = "<< *ptr;
	return 0;
	}
		#include <iostream>
		using namespace std;
		int main()
		{
		int var1 = 0,var2 = 0;
		const int* const ptr = &var1;
		*ptr = 1;
		ptr = &var2;
		cout<<"Var1 = "<<*ptr;
		return 0;
		}

9. Write a program to enter a size of array. Create an array of size given by user **using** "**new" Dynamic memory management operator (free store operator).** Enter the data to store in array and display the data after adding 2 to each element in the array. Delete the array by **using "delete" memory management operator.**

10. Find the output of following program. Explain the **use of bool data type**.

```
#include<iostream>
using namespace std;
int main()
{       bool a = 321, b;
        cout << "Bool a Contains : " << a<<endl;
        int c = true; int d = false;
        cout<<"c = "<<c <<endl<<"d = "<<d;
        c = a + a;
        cout << "\nInteger c contain : " << c;
        b = c + a;
        cout << "\nBool b contain : " <<b;
        return 0;}
```

Function

11. Define three functions named divide(). First function takes numerator and denominator as an input argument and checks it‟s divisible or not, Second function takes one int number as input argument and checks whether the number is prime or not and Third function takes 3 float number as arguments, divide it with 3 and print average of the numbers Use concept of **Function Overloading / static binding.**

12. Define four function void swap () which accepts two arguments by reference and swap the values. First function swaps two characters, second function swaps two integers, third function swaps two floats values and fourth function swaps two double values. Use the concept of **call by reference** in all four functions and **function overloading and inline function.**

13. Write a function called tonLarge( ) that takes two integer arguments by **reference** and then sets the larger of the two numbers to 100 using **Return by reference**. Write a main( ) program to exercise this function.

14. Write a function called power ( ) that takes two arguments: a double value for **n** and an int for **p**, and returns the result as double value. Use **default argument** of 2 for p, so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function.

Classes and Objects

15. Define a **C++ Structure** Rectangle with data member‟s width and height. It has get_values( ) member functions to get the data from user and area( ) member functions to print the area of rectangle. Also create a **C++ Class** for the above program. **Define both functions inside the class. Member function defined inside the class behaves like an inline function** and illustrate the difference between C++ Structure and C++ Class.

16. Write a C++ program having class **Batsman.** It has private data members: batsman name, bcode (4 Digit Code Number), innings, not out, runs, batting average. Innings, not out and runs are in integer and batting average is in float.
    Define following **function outside the class using scope resolution operator.**
    1) Public member function getdata()to read values of data members.
    2) Public member function putdata()to display values of data members.
    3) Private member function calcavg() which calculates the batting average of a batsman. Also make this outside function **inline.**
    Hint : batting average = runs/(innings - notout)

17. Define class Digit having int „n‟ as data member. Define member function enter() to enter the data and show() to print the data. A class has member function compare() that displays whether the first object is smaller, greater or same as compared to second object. (Function compare() should support: int x = d1.compare(d2); where d1 and d2 are objects of class Digit). **Use Concept of Object as Function Arguments.**

18. Define class *Currency* having two integer data members rupee and paisa. A class has member functions enter() to get the data and show() to print the amount in 22.50 format. Define one member function *sum*() that adds two objects of the class and stores answer in the third object i.e. c3=c1.sum (c2). The second member function should add two objects of type currency passed as arguments such that it supports c3.add(c1,c2); where c1, c2 and c3 are objects of class *Currency*. Also Validate your answer if paisa >100. Write a main( )program to test all the functions. **Use concepts of Object as Function Arguments, function returning object and function overloading.**

19. Define a class **Dist** with int feet and float inches. Define member function that displays distance in 1‟-2.5” format. Also define member function scale ( ) function that takes **object by reference** and scale factor in float as an input argument. The function will scale the distance accordingly.
For example, 20‟-5.5” and Scale Factor is 0.5 then answer is 10‟-2.75”

20. Create a Class Gate for students appearing in Gate (Graduate Aptitude test for Engineering) exam. There are three examination center Vadodara, Surat, and Ahmedabad where Gate exams are conducted. A class has data members: Registration number, Name of student, Examination center. Class also Contains static data member ECV_Cnt, ECS_Cnt and ECA_Cnt which counts the number of students in Vadodara, Surat and Ahmedabad exam center respectively. Class Contains two Member function getdata () which gets all information of students and counts total students in each exam center and pudata () which prints all information about the students. Class also contains one static member function getcount () which displays the total number of students in each examination center. Write a program for 5 students and display the total number of students in each examination center. **Use static data member, static member function and Array of Objects.**

21. Define a class *Fahrenheit* with float temp as data member. Define another class *Celsius* with float temperature as data member. Both classes have member functions to input and print data. Write a non-member function that receives objects of both the classes and declare which one is higher than another according to their values. Also define main() to test the function. Define all member functions outside the class. (Formula for converting Celsius to Fahrenheit is F = (9C/5) + 32). **Use the concept of friend function.**

22. Create a Class Date having data members: int dd, mm, yyyy. Class has one member function to input the dates and another member function which prints the dates. Write a main() function which takes two dates as input. Write a friend function swapdates() which takes two objects by reference of type Date and swaps both the dates. **Use the concept of Friend function which takes objects by reference**

23. Create a class Customer having data members: name of the customer and customer number in integer and member function to get customer data. Create another class Manager having data members: name of manager and employee id in integer and member function to get managers data. Class Manager also have member function get_cust_data () which takes objects of class Customer as input and prints the customers details and is a friend function of class Customer . Write a main () function to test all this function. Use the concepts of **Member function of one class can be a Friend Function of another class.**

24. Create a class Child having data members: name of the child and gender and a member function to get and print child data. Create another class Parent which is a friend class of child class. Class Parent have member function ReadChildData() which takes child‟s object by reference as input argument and Reads the childs data and DisplayChildData() which takes childs object as argument and displays childs data. Use the concepts of **Friend Class.**

25. Check the following C++ code and find if there is any error in code, give justification for the error, correct the code and write the output:

1. Example of const member functions

```
#include<iostream>

using namespace std;
class sample
{
  int m, n;
 public:
 void getdata();
 void putdata() const;
};
void sample::getdata()
{
 cout<< "Enter m & n";
cin>>m>>n;
}
void sample::putdata() const
```

```
{
 m=12;
 n=34;
 cout<< " m = "<<m<<"n= "<<n;
}
int main()
{
 sample s1;
 s1.getdata();
 s1.putdata();
 return 0;
}
```

2. Example of (a)Pointer to data members, (b)Pointer to member functions

(a)

```
#include<iostream>
using namespace std;
class student
{
  public: int roll_no;
};
int main()
{
// declaring pointer to data member
        int student :: *p1 = &student::roll_no;
        student s;
        student *optr = &s;
        s->*p1 = 42;
        cout<<"Roll no is "<<s->*p1<<endl;
        optr.*p1 = 45;
        cout<<"Roll no is"<<optr.*p1<<endl;
        return 0;
}
```

(b)

```
#include<iostream>
class employee
{
public:
   void hello()
   {
```

```
        cout<<"Hi hello"<<endl;
    }
};
int main()
{
    // declaring pointer to member function hello
    void (employee ::*fp)() = &employee::hello;
    employee e;
    employee *optr = &e;
    (e->*fp)();
    (optr.*fp)();
    return 0;
}
```

3. Example of Local Classes

```
#include<iostream>
using namespace std;
void testlocalclass()
{
        class Test
        {
        static int cnt;
        public:
        void set()
        {cout<<"Enter Count: "; cin>>cnt; }
        void get();
        };
        void Test:: get()
        { cout<<"Count: = " <<cnt; }
        Test t;
        t.set();
        t.get();
}
int main()
{
testlocalclass();
    return 0;
}
```

Constructor and Destructor

26. Write a C++ program having class **time** with data members: hr, min and sec. Define following member functions.
    1) getdata() to enter hour, minute and second values
    2) putdata() to print the time in the format 11:59:59
    3) default constructor
    4) parameterized constructor
    5) copy constructor
    6) Destructor.

    Use 52 as default value for sec in parameterized constructor.
    Use the concepts of default constructor, parameterized constructor, Copy constructor, constructor with default arguments and destructor.

Operator Overloading & Type Conversion

27. Create a class Number having int num as member. The class has input and output functions. Overload unary operator (++) such that it supports N1=N2++ and N3=++N1 and Overload unary (-) such that it supports

    N3 = - N3. Also define default, parameterized and copy constructor for the class. Also explain use of nameless object in operator overloading. **Use the concept of Overloading Unary Operators. Operator overloading is also known as Compile Time Polymorphism or static binding.**

28. Create a class complex having data members int real , img and member function to print data. Overload Unary operator (-) using friend function such that it supports –C1 where C1 is the object of class complex. Also define default, parameterized and copy constructor for the class.
    Use the concept of Overloading Unary Operators with friend function.

29. Create a class String having character array. Class includes constructor and required member functions to get and display the object. Overload the operators +(s3=s1+s2), ==(s1<s2), +=(s1+=s2) for the class. **Use the concept of Overloading Binary Operators**

30. Create a class Measure having members: meter and cm. The class has get( ) and put( ) functions. Overload operator + and – such that they support M1=M2+15 and M3=M1 – 4.5. Also overload + and – such that they support M1=5.0+M2 and M3=2.0 – M4. Write a main( ) to test the class.
    Use the concept of Overloading Binary Operators with friend function.

31. Create a class Celsius with float. Define appropriate member functions such that it support the statements: C1=30.5F; float temperature; temperature=C2;
Use the concept of **Type conversion from basic type to class type and class type to basic type.**

32. Create classes Celsius and Fahrenheit with float. Define appropriate member functions such that they support the statements in main( ): Celsius C1, C2=5.0; Fahrenheit F1, F2; F1=C2; C1=F2;
Use the concepts of **Type conversion from class type to class type.**
Write this Program in two ways.
Define appropriate member function in class Celsius**.**
Define appropriate member function in class Fahrenheit.

Inheritance

34. Define a Base Class Vegetable having data member Color and member function getdata() which takes color as an input and putdata() which print the color as an output.
Vegetable Class has one subclass named Tomato having data members weight and size and member function gtdata() which takes weight and size as an input and ptdata() which prints weight and size as output. Write a C++ Program which inherits the data of Vegetable class in Tomato class using **Single Inheritance.**

35. Write a program to create a class **Medicine** which stores type of medicine, name of company, date of manufacturing. Class **Tablet** is inherited from Medicine. Tablet class has name of tablet, quantity per pack, price of one tablet as members. Class **Syrup** is also inherited from Medicine and it has quantity per bottle, dosage unit as members. Both the classes contain necessary member functions for input and output data. Write a main( ) that enter data for tablet and syrup, also display the data. **Use the concepts of Hierarchical Inheritance.**

36. Create a class shape having data member shape_name and member function to get and print shape_name. Derive a Class Circle which is inherited publicly from class shape and having data members radius of a circle and member function to get and print radius of a circle. Derive a Class Area which is inherited publicly from Class Circle and having data members area_of_circle and member function display () which displays area of a circle. Use object of class Area in main () function and get and display all the information. **Use the concepts of Multilevel Inheritance.**

37. Define a class Hospital having rollno and name as data members and member function to get and print data. Derive a class Ward from class Hospital having data members: ward number and member function to get and print data. Derive another class Room from Hospital having data member bed number and nature of illness and member function to get and print data. Derive class Patient from Class Ward and Class Room. In main () declare 5 object of Class Patient and get and display all the information. **Use the concept of Virtual Base Class and Hybrid Inheritance.**

38. Create a Class alpha having data member: int x and one argument constructor which initializes the value of x. It also has member function which displays the value of x. Create another class beta which contains data member: float y and one argument constructor which initializes the value of y. It also has member function which displays the value of y. Create a Class Gamma which publicly inherits from class alpha and class beta and has two data members: int m, n and a constructor which passes argument to the base class constructor as well as initializes its own data members. Class Gamma also has member function to print the values of m and n. Write main function which creates object of class Gamma which passes values of base class constructor as well as derived class constructor. **Use the concept of Multiple Inheritance and Constructor in Derived Class.**

39. Create one application n a group of 3 person which implement all type of inheritance

Pointers and Virtual Functions

40. What is the output of the following code:

(a) Pointer to Objects

```
#include<iostream>
using namespace std;
class product
{
        int code;
        float price;
public:
        void getdata(int a, float b)
        {
    code=a;
    price=b;
        }
        void show()
        {
    cout<<"Code: "<<code<<endl;
```

```
        cout<<"Price: "<<price<<endl;
          }
};
int main()
{
   product * p = new product;
   product *d = p;

          int x,i;
          float y;
   cout<<"Input code and price for product: ";
   cin>>x>>y;

   p->getdata(x,y);

   d->show();
}
```

(b) **this pointer**

```
#include<iostream>
using namespace std;
class student
{
          int roll_no;
          float age;
public:
          student(int r, float a)
          {
          roll_no = r;
          age = a;
          }
          student & greater (student & x)
          {
          if(x.age>=age)
          return x;
          else
          return *this;
          }
          void display()
          {
          cout<<"Roll No "<<roll_no<<endl;
```

```
        cout<<"Age "<<age<<endl;
        }
};
int main()
{
        student s1 (23,18),s2 (30,20),s3 (45,16);
        student s = s1.greater(s3);
        cout<<"Elder Person is :"<<endl;
        s.display();

}
```

(c) Pointers to Derived Objects

```
#include<iostream>
using namespace std;
class BC
{
 public:
        int b;
        void show()
        {
        cout<<"b = "<<b<<endl;
        }
};
class DC : public BC
{
public:
        int d;
        void show()
        {
        cout<<"b = "<<b<<endl;
        cout<<"d = "<<d<<endl;
        }
};
int main()
{
        BC *bptr;
        BC base;
        bptr = &base;
        bptr->b = 100;
        cout<<"bptr poins to base objects"<<endl;
```

```
        bptr->show();
        DC derived;
        bptr = &derived;
        bptr->b = 200;

        /*bptr->b = 300;*/ // wont work
        cout<<"bptr now points to derived object"<<endl;
        bptr->show();
        DC *dptr;
        dptr=&derived;
        dptr->d=300;
        cout<<"Dptr is derived type pointer"<<endl;
        dptr->show();
        return 0;
}
```

41. Create a class Media that stores the title (a string) and price (float). Class Media has two argument constructor which initializes data members of class Media. Also declare a virtual function display () in Class Media. From the class Media derive two classes: Class book, which contains data member page count (int): and Class tape, which contains data member playing time in minutes (float). Both Class book and Class tape should have a constructor which initializes base class constructor as well as its own data members and display ( ) function which displays book details and tape details respectively. Write a main ( ) to test book and tape classes by creating instances of them, asking the user to fill data and displaying them. **Use the concept of Virtual function and Constructor in Derived Class. Virtual function is also known as Runtime Polymorphism or Dynamic Binding.**

42. Create a Abstract class vehicle having average as data and pure virtual function getdata() and putdata(). Derive class car and truck from class vehicle having data members: fuel type (petrol, diesel, CNG) and no of wheels respectively. Write a main ( ) that enters the data of two cars and a truck and display the details of them. **Use the concept of Abstract Base class and Pure Virtual functions.**

File Handling

43. Write a program that creates a text file that contains ABC…Z. A program should print the file in reverse order on the screen. i.e. ZYX…BA. Use concept of**Opening the file using constructor and open () function. Use all error handling functions like eof() , fail() , bad() , good() and functions for manipulation of file pointer like seekg() and tellg().**