SVKM'S NMIMS

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT& ENGINEERING

Academic Year: 2020-2021

Program: B Tech Stream : Computer Science & Business Systems Year: II Semester: III

Subject: Object Oriented Programming Time: 2 Hrs.

Date: 24th Nov 2020 No. of Pages: 3

Marks: 50

Final Examination

Instructions: Candidates should read carefully the instructions printed on the question paper

1. Question 1 is compulsory.
2. Solve any four out of remaining five questions
3. All questions carry equal marks.
4. Answer to each new question to be started on a fresh page.
5. Figures in brackets on the right hand side indicate full marks.
6. Assume Suitable data if necessary.

Q1. a.	Explain Function call by reference with an example.	[2]
Q1. b.	In Object Oriented Paradigm how Access specifiers are used to implement data hiding. Demonstrate with an example.	[3]
Q1. c.	Write a program to overload the unary ++ operator to perform increment operation on given objects.	[3]
Q1. d.	Design a Website management or administration UML use case diagrams for given scenario. The Website Administrator could manage user groups, users, user sessions, and logs. The Help Desk staff should be able to only manage the users.	[2]
Q2. a.	Write a function that takes two Distance values as arguments and returns the larger one. Create a structure to store the distance values in a structure. Include a main() program that accepts two Distance values from the user in feets and inches, compares them, and displays the larger. The program should also validate that the value of inches should always be less than 12 inches.	[6]
Q2. b.	Create a class that imitates part of the functionality of the basic data type int. Call the class Int (note different capitalization). The only data in this class is an int variable. Include member functions to initialize an Int, to initialize it to an int value, to display it, and to add two Int objects. Write a program that exercises this class by creating one uninitialized and two initialized Int values, adding the two initialized values and placing the response in the uninitialized value, and then displaying this result.	[4]
Q3. a.	Explain the concept of new and delete operators with a suitable example.	[4]

Q3. b.	int main()
	{
	Mystring a;
	char temp[] = "Hello";
	Mystring b{ temp };
	Mystring c{ b };
	c.display();
		[6]

```
  ~c;
}
```

Write a suitable class mystring with appropriate function to execute the above main().

Q4. a. Write a program that consists of two base classes and one derived class The base class basic_info contains the data members: name, rollnumber, sex. An another base class academic_fit contains the data members: course, semester and rank. The derived class (derived from both the base classes) financial_assit contains the data member amount besides the data members of the base classes. The derived class has been declared as public inheritance. The member functions are used to get information of the derived class from the keyboard and display the contents of class objects on the screen. [5]

Q4. b. Justify the use of Friend class with an example. [5]

Q5. a. Design a template class 'myIncrement' which has a constructor to initialize a value and a template function toIncrement() that increments the value by 1. This particular class will work perfectly for all the data types except for char. So, instead of incrementing the value for char, give it a special behavior and convert the character to uppercase. Write a main function and pass integer, double and character values to function. In output the number type values should be incremented by 1 and characters should be printed in uppercase. [6]

Q5. b. Observe the program segment given below carefully and fill the blanks marked as Line 1 and Line 2 using fstream functions for performing the required task. [4]

```
#include <fstream.h>
class Library
{
 long Ano; //Ano - Accession Number of the Book
char Title[20]; //Title - Title of the Book
int Qty; //Qty - Number of Books in Library
public:
void Enter (int); //Function to enter the content
void Display(); //Function to display the content
void Buy(int Tqty)
{
Qty+=Tqty;
} //Function to increment in Qty
long GetAno( )
{
return Ano;
}
};
void BuyBook(long BANo,int BQty) //BANo -
//BQty -
{ fstream File;
File.open("STOCK.DAT" ,ios::binary|ios::in|ios::out);
int position=-l;
Library L;
while(Position==-l &&File.read((char*)&L,sizeof(L)))
if (L.GetAno()==BANo)
{
L.Buy(BQty);
//To update the number of Books
Position = File.tellg()-sizeof(L) ;

_____;
```

```
//Line 1: To place the file pointer to the required position

_____;
//Line 2:To write the object L on to the binary file

}
if (Position==-l)
cout<< "No updation do:r{e as required Ano not found..";
File.close( );
}
```

Q6. a. Identify the exceptions in the given code and write the corrected code with exceptions handling. [4]

```
#include <iostream>
using namespace std;
void fun(int *ptr, int x)
{
   if (ptr == NULL)
      throw ptr;
   if (x == 0)
      throw x;
   /* Some functionality */
}
int main()
{
   fun(NULL, 0);
   return 0;
}
```

Q6. b. Design a UML class diagram which shows a domain model for online shopping. [6]

In the given system customer has unique id and is linked to exactly one account. Account owns shopping cart and orders. Customer could register as a web user to be able to buy items online. Customer is not required to be a web user because purchases could also be made by phone or by ordering from catalogues. Web user has login name which also serves as unique id. Web user could be in several states - new, active, temporary blocked, or banned, and be linked to a shopping cart. Shopping cart belongs to account. Account owns customer orders. Customer may have no orders. Customer orders are sorted and unique. Each order could refer to several **payments**, possibly none. Every payment has unique id and is related to exactly one account. Each order has current order status. Both order and shopping cart have **line items** linked to a specific product. Each line item is related to exactly one product. A product could be associated to many line items or no item at all.