Code: AC55/AT55   **Subject: OBJECT ORIENTED PROGRAMMING WITH C++**

AMIETE – CS/IT (NEW SCHEME)

Time: 3 Hours | **JUNE 2012** | **Max. Marks: 100**

PLEASE WRITE YOUR ROLL NO. AT THE SPACE PROVIDED ON EACH PAGE IMMEDIATELY AFTER RECEIVING THE QUESTION PAPER.

NOTE: There are 9 Questions in all.
- **Question 1 is compulsory and carries 20 marks. Answer to Q.1 must be written in the space provided for it in the answer book supplied and nowhere else.**
- **The answer sheet for the Q.1 will be collected by the invigilator after 45 Minutes of the commencement of the examination.**
- **Out of the remaining EIGHT Questions answer any FIVE Questions. Each question carries 16 marks.**
- **Any required data not explicitly given, may be suitably assumed and stated.**

Q.1   **Choose the correct or the best alternative in the following:**   **(2×10)**

a. Which of the following operators below allow to define the member functions of a class outside the class?

   (A) ::   **(B)** ?
   (C) :?   **(D)** %

b. How do we define a constructor?

   (A) X~() {}   **(B)** X() {}~
   (C) X() ~{}   **(D)** ~X() {}

c. What is the output of the following code snippet assuming user enters the side as 4?

```
         class square
        {
                public:
                    double side1;
                    double area()
                    {
                            return(side1*side1);
                    }
         };
        int main()
        {
            double area1=0;
            square c1,c2;
            cout << "Enter the length of the square" << endl;
            cin >> c1.side;
            cout << "The area of the square is : " << c1.area1() << endl;
            return(0);
        }
```

   (A) 16   **(B)** Will result in an error
   (C) 8   **(D)** 12

d. C++ provides facility to specify that the compiler should match function calls with the correct definition at the run time. This process is called as

(A) Static binding   (B) Dynamic Binding
(C) Inline functions   (D) Friend functions

e. Which type of casting can be used only with pointers and references to objects?

(A) dynamic_cast   (B) cast
(C) static_cast   (D) pointer_Cast

f. ios::ate is used for

(A) Set the initial position at the end of the file
(B) Set the initial position at the start of the file
(C) Set the last position at the end of the file
(D) Set the last position at the start of the file

g. int a = 10;
   void main()
   {
                int a = 20;
                cout << a << ::a;
    }
      The output of this program is

(A) Syntax error   (B) 10 20
(C) 20 10   (D) 20 20

h. When class B is inherited from class A, what is the order in which the constructers of those classes are called?

(A) Class A first Class B next   (B) Class B first Class A next
(C) Class B's only as it is the child class (D) Class A's only as it is the parent class

i. Write the output of the following:
char *mychar; //points to memory location 1000

short *myshort; //points to memory location 2000

long *mylong //points to memory location 3000

mychar++;

++myshort;

mylong++;

cout << mychar << myshort << mylong;

(A) 1001 2001 3001   (B) 1001 2002 3004
(C) 1001 2001 3002   (D) 1001 2002 3000

j.   Which of the following is not a standard exception built in C++?

    (A) std::bad_creat   **(B)** std::bad_alloc
    (C) std::bad_cast   **(D)** std::bad_typeid

Answer any FIVE Questions out of EIGHT Questions.
Each question carries 16 marks.

Q.2 a. Explain three main characteristics of OOP. **(6)**

    b. What is dynamic memory management? How is it handled in C++? Give examples. **(6)**

    c. Explain volatile and const qualifiers in C++. **(4)**

Q.3 a. Briefly explain the four different jump statements. **(8)**

    b. Explain with the help of examples, differences between arrays and pointers. **(4)**

    c. What are nested structures? Give example. **(4)**

Q.4 a. What is parameter passing? Explain the various parameter passing schemes supported by C++. **(8)**

    b. Illustrate through suitable examples, the difference between function declaration, function definition and function call. **(8)**

Q.5 a. Define class and object. Write a class "Rectangle" containing two data items "length" and 'breadth" and four functions setData() , getData(), displayData() and area() to set the length and breadth, to get the user inputs, to display and to find the area of the rectangle respectively. Also write a program which declares the objects and uses the member functions of the class. **(6)**

    b. Write a C++ program to create class called STACK using array of integers. Implement the push(), pop(), isfull() and isempty() functions. Also display the status and contents of the stack after each operation. **(10)**

Q.6 a. Describe operator overloading. Why is it necessary? Explain the restrictions of overloading of an operator with an example. **(6)**

    b. Define a Fraction class having a numerator and denominator. Write the following member functions for the class
      (i)   a constructor to initialize the members.
      (ii) overload ++ operator (both prefix and postfix forms) **(10)**

Q.7 a. What is inheritance? Explain different types of inheritance with suitable diagram and syntax. **(8)**

b. What is runtime polymorphism? How virtual functions can be used to implement the runtime polymorphism? Explain with an example.   **(8)**

Q.8 a. What is exception handling? Mention three keywords that are provided in C++ for implementing exception handling.   **(6)**

b. With syntax, explain class templates. Write a C++ program to demonstrate usage of function templates with multiple parameters.   **(6)**

c. Write a C++ program to create class template to represent generic vector. Include member functions to
   (i) Create a vector
   (ii) To display the contents of a vector   **(4)**

Q.9 a. Explain the concept of iostreams provided in C++. Explain in detail stream class hierarchy.   **(8)**

b. Write a C++ program to copy one file to another file after converting the lower case characters to upper case characters.   **(8)**