Reg. No. : | | | | | | | | | | | |

K 3530

M.C.A. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2007.

Second Semester

MC 1652 — OBJECT ORIENTED PROGRAMMING

(Regulation 2005)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is procedural programming?

2. Define an Object.

3. What is an inline function?

4. What is a reference variable?

5. What is the purpose of constructor and destructor?

6. What is function overloading?

7. What is reusability?

8. Define template and list the different types of template.

9. What is an interface?

10. What is the significance of abstract type?

PART B — (5 × 16 = 80 marks)

11. (a) Discuss the advantages, limitations of procedural oriented programming and object oriented programming with suitable examples.

Or

(b) Discuss the following terms with suitable illustration :

(i) Objects

(ii) Data abstraction and encapsulation

(iii) Inheritance

(iv) Polymorphism. $(4 \times 4 = 16)$

12. (a) Define a class complex with two data members real and imaginary and three member functions for

(i) Input (4)

(ii) Displaying the object complex (4)

(iii) Subtracting two objects of class complex. Test the class using main ( ). (8)

Or

(b) Create a class called time that has separate data members for hours, minutes and seconds. Use three member functions for

(i) reading data (4)

(ii) displaying the object time (4)

(iii) adding two objects of class time test the class using main ( ). (8)

13. (a) What is operator overloading? Develop a C++ program to overload unary operator for processing the objects of a class called counter.

Or

(b) With relevant examples discuss the different types of constructors in C++.

14. (a) What is inheritance? With relevant examples discuss the different types of inheritance in C++.

Or

(b) Justify the need for virtual functions in C++. Explain how dynamic binding is achieved in C++ with a program segment.

15. (a) Explain the role and different types of classes with an example.

Or

(b) Write short notes on :

(i) Interface classes (6)

(ii) I/O operations using streams. (10)
