

**B.TECH / CSE /3<sup>RD</sup> SEM/ CSEN 2103/2017**  
**OBJECT ORIENTED PROGRAMMING**  
**(CSEN 2103)**

**Time Allotted : 3 hrs**

**Full Marks : 70**

*Figures out of the right margin indicate full marks.*

*Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.*

*Candidates are required to give answer in their own words as far as practicable.*

**Group - A**

**(Multiple Choice Type Questions)**

1. Choose the correct alternative for the following: **10 × 1 = 10**
- (i) Which of the following is not supported in JAVA?  
(a) Single Inheritance (b) Multilevel Inheritance  
(c) Multiple Inheritance (b) None of these.
  - (ii) A subclass of an abstract class will be abstract  
(a) always (b) never  
(c) when abstract super class is partially implemented  
(d) none of these.
  - (iii) Which one is weaker access specifier in Java?  
(a) protected (b) default  
(c) private (b) public.
  - (iv) Which of the following cannot be instantiated?  
(a) Abstract Class (b) Concrete class  
(c) Interface (b) Exception class.
  - (v) Which one is not a Thread class method?  
(a) init (b) paint (c) start (d)run.
  - (vi) Which of the following permits function overloading on c++?  
(a) type (b) number of arguments  
(c) both of the mentioned (b) none of the mentioned.
  - (vii) Overridden methods are an example of  
(a) Abstraction (b) Encapsulation  
(c) Polymorphism (b) None of the above.

**B.TECH / CSE/3<sup>RD</sup> SEM/ CSEN 2103/2017**

- (viii) What will happen while using pass by reference?  
(a) The values of those variables are passed to the function so that it can manipulate them  
(b) The location of variable in memory is passed to the function so that it can use the same memory area for its processing  
(c) The function declaration should contain ampersand (& in its type declaration)  
(d) All of the mentioned.
- (ix) Which is used to create a pure virtual function ?  
(a) \$ (b) =0 (c) & (d)!
- (x) Which keyword can be used in template?  
(a) class (b) type name  
(c) both a & b (b) function.

**Group - B**

2. (a) `int sum(int a=5, int b=9) {...}`  
What is the advantage of having a function declaration of the above type? Explain by completing the example.
- (b) `int sum(int n,...)`  
What do the dots in the above function call imply and when are function declarations of this type used? Give a list of macros associated with such function calls and their usage.
- (c) 

```
#include<iostream>
using namespace std;
class Myclass {
int x, y;
public:
    Myclass(): x(5),y(5) {}
    Myclass(int a=0, int b=0) {x=a;y=b;}
void sum() {cout<<x+y;}
};
int main() {
    Myclass obj;
    obj.sum();
}
```

The above code will not compile. Identify the error and explain why. Give the corrected code.

**4+ 4 + 4 = 12**

3. (a) "Friendship between two classes is not mutual" – portray the same with a short example.

(b) What is the scope of a static object of a class?

```
(c) #include<iostream>
using namespace std;
class Myclass {
public:
    Myclass() { cout << "Creating object "<<endl; }
    ~Myclass() { cout<< "Deleting object " <<endl; }
    void Myfunction() { Myclass static obj; }
};
int main() {
    Myclass obj1;
    obj1.Myfunction();
    cout <<"Exiting main"<<endl;
}
```

What will be the output of the above program. Mention the object name against the creation and deletion statements to identify which object they refer to.

(d) What is the difference between a constructor and a destructor?

(e) What is the default access specifier of class data members and how does that promote encapsulation?

$$3 + 2 + 3 + 2 + 2 = 12$$

#### Group - C

4. (a) class student {...};  
class study:public student{...};  
class sport:public student{...};  
class result:public study,public sport{...};

In the above pseudo-code, the sequence of inheritance causes a certain type of compilation issue. Identify the issue and correct the pseudo-code mentioning the solution.

(b) What are the different types of inheritance? Draw a diagram for each type and explain.

```
(c) #include<iostream>
class father {
public: void showAge() {
    cout << "Father's age 55"<<endl;
} };
class son:public father {
public: void showAge() {
    cout << "Son's age 20"<<endl;
} };
int main() {
    father *fptr; fptr = new father();
    fptr->showAge();
    fptr=new son(); fptr->showAge();
    delete fptr; }
```

What will be the output of the above program? Taking this program as example, explain what a virtual function is and how virtual function can be used to change the behaviour of the above program.

$$4 + 4 + 4 = 12$$

5. (a) What are the applications of *this* pointer? Why do we need virtual functions? What is an abstract class in C++?

(b) What is the use of throw keyword in C++? Explain with example. What is the use of template and namespace in C++?

$$(2 + 2 + 1) + (3 + 2 + 2) = 12$$

#### Group - D

6. (a) What do you understand by platform-independence of JAVA? Explain constructor overloading in JAVA with an example. How does protected and default access specifiers differ?

```
(b) class Outer {
    int outer_x = 100;
    void test() { System.out.println("outer x and inner y= " + outer_x + y); }
    class Inner {
        static int y = 10;
        void display() { System.out.println("outer x and inner y= " + outer_x + y); }
    }
    class Myclass {
        public static void main(String args[]) {
        }
    }
}
```

The above code has two errors. What are they?

- (c) Correct the above code. Instantiate class Inner directly from main in the above code. Then call display() on the object.

$$(1 + 3 + 1) + 3 = 12$$

7. (a) Class RBI has atleast one method interestRate(). Class HDFC and PNB inherit class RBI. Since they each have their own interest rates they have their own implementation of the method interestRate(). Write a code in java for the scenario and point out the instances of runtime polymorphism.

- (b) Why are abstract classes needed? What is the need of abstract methods in abstract class? Can abstract class have objects?

- (c) Give a suitable abstract class as the common baser-class of the classes mentioned in Question in Q-7-a, where only one abstract method "interestRate" is declared. How can run-time polymorphism be achieved (illustrate using example) using pointer(s) to the abstract class.

$$3 + (2 + 2 + 1) + (2 + 2) = 12$$

#### Group - E

8. (a) What is Java's solution to multiple inheritance?

- (b) interface vehicle {  
void caution(){ System.out.println("Drive safe! "); }  
void speed();  
}

What is wrong in the above interface? How can you rectify the mistake without changing any of the method implementations?

- (c) Class car implements the above (corrected) interface. Access the methods speed and caution using interface reference variable.

- (d) What are the cases when an interface can implement a method body?

$$2 + 3 + 4 + 3 = 12$$

9. (a) What do you mean by context switch with reference to java threads and what are the rules that determine when context switch will take place?

- (b) What are the two approaches of creating a thread in java? Which approach do you prefer and why?

- (c) Mention the different states a thread can be in and give a brief description for each.

- (d) How can you obtain the state of a thread?

$$3 + 3 + 4 + 2 = 12$$