**Code: AC55/AT55/ AC105/AT105**

**Subject: OBJECT ORIENTED PROGRAMMING WITH C++**

## AMIETE – CS/IT (Current & New Scheme)

**Time: 3 Hours**     **December - 2017**     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.   _____ is used to convert the value of one type to another.
   **(A)** Data type          **(B)** Variable
   **(C)** Typecasting        **(D)** None of these

b.   The process in which the code to be linked with the procedure call at compile time is called
   **(A)** binding            **(B)** late binding
   **(C)** static binding     **(D)** dynamic binding

c.   A template can be used to create a family of
   **(A)** classes            **(B)** functions
   **(C)** classes and Functions   **(D)** None of these

d.   _____allows us to create new classes based on existing classes.
   **(A)** Copy construction  **(B)** Function overloading
   **(C)** Inheritance        **(D)** Polymorphism

e.   Which of the following statements regarding inline functions is incorrect?
   **(A)** It speeds up the execution   **(B)** It increases the code size
   **(C)** It slows down the execution  **(D)** All of these

f.   For proper display of polymorphism, a method in the base class that must be declared for the purpose of display of polymorphism is known as _____.
   **(A)** private            **(B)** public
   **(C)** protected          **(D)** virtual

g.   The operator that cannot be overloaded is
   **(A)** increment          **(B)** scope resolution
   **(C)** decrement          **(D)** None of these

h.   If a file is created by 'ifstream', then the default mode of the file is
   **(A)** ios :: ate         **(B)** ios :: in
   **(C)** ios :: app         **(D)** ios :: nocreate

**Code: AC55/AT55/ AC105/AT105**
**Subject: OBJECT ORIENTED PROGRAMMING WITH C++**

   i. What will be the output of following program?

```
#include<iostream.h>
void main()
{
float x;
x=(int)9/2;
cout<<x;
}
```

   **(A)** 4.5                    **(B)** 4.0
   **(C)** 4                      **(D)** 5

   j. When an exception is thrown, additional information sent may be placed in
      **(A)** the throw keyword
      **(B)** the catch block
      **(C)** the try keyword
      **(D)** an object of the exception class

---

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

---

**Q.2**  a.  Define Programming paradigms. Explain procedural and functional programming paradigms with suitable example.   **(3+4)**

   b.  In the context of OOP, define the following terms.   **(5)**
       (i) Abstration and Encapsulation
       (ii) static and dynamic binding

   c.  What do you mean by lvalue? What will be the output of following program?   **(4)**
```
# include<iostream.h>
Int  main()
{
int a=12;
int b=6;
int c=18;
cout<< "a=" << a<< ", b= " << b << " , c = " << c << end l;
++c = (a++) + (b++);
cout<< "a=" << a<< ", b= " << b << " , c = " << c << end l;
return 0;
}
```

**Q.3**  a.  Define the following statements in object oriented programming with suitable example to illustrate.   **(3×3)**
       (i) Expression Statement          (ii) Compound Statement
       (iii) Jump Statement

   b.  Describe the address of operator. What will be the output of following program?   **(7)**
```
# include<iostream.h>
int main()
{
int i=512;
char *c=(char*)&i;
```

```
    c[0]=1;
cout<<"i is:"<<i<<endl;

return 0;
}
```

**Q.4** a. Define and explain the arguments passed by value and by reference. **(5)**

b. Describe the Inline function. When do we use this function? Illustrate the function by writing a C++ program. **(6)**

c. Explain the concept of scope of variable. Use the example to illustrate it. **(5)**

**Q.5** a. Define the constructor and destructor member functions. What are the constraints that must be followed during the declaration of a constructor and destructor. **(6)**

b. Write a program in C++ to find the factorial of a number using a constructor. **(5)**

c. Write short notes on the following: **(5)**
(i) Static class and function members
(ii) New and Delete Operators

**Q.6** a. Write a program to add two complex numbers using binary operator overloading. **(6)**

b. Define the cast operator. Write a program to illustrate. **(6)**

c. The new and delete operator is used as a way of allocating memory. Why do we overload new and delete operator? **(4)**

**Q.7** a. Define the term inheritance. What are the benefits and limitations of inheritance? **(4)**

b. Describe the multiple inheritance write a C++ program to illustrate it. **(6)**

c. Define polymorphism. Explain the compile time and run time polymorphism. Give the C++ program to illustrate the concept of compile time and run time polymorphism. **(6)**

**Q.8** a. Define Multiple Catch statement. Describe the role of try, throw and catch in Multiple Catch. Write a program in C++ for Exception Handling with Multiple Catch. **(8)**

b. Describe the concept of template in object oriented programming. Explain the class template and function template with suitable example in C++. **(8)**

**Q.9** a. Define a file. What is the roll of input and output stream in file? Explain the Stream classes for file operations, in detail. **(8)**

b. What is a file mode? Describe the various file mode options available, in detail. **(8)**