This question paper contains **10** printed pages]

Roll No. | | | | | | | | | | |

S. No. of Question Paper : **3527**

Unique Paper Code : **2344001101**

Name of the Paper : **GE : Programming Using C++**

Name of the Course : **B.Sc.(P)/B.Sc. Maths. Sc.**

Semester : **I**

Duration : **3 Hours** Maximum Marks : **90**

(Write your Roll No. on the top immediately on receipt of this question paper.)

Paper has two sections. *All* the questions in **Section A** are compulsory.

Answer any *four* questions from **Section B.**

Parts of question must be answered together.

Section A

(Compulsory)

1. (*a*) Give the C++ syntax for the following : (

(*i*) A `constant float` with the value 3.14.

(*ii*) The function prototype of an `inline` function `F1` that accepts two parameters and returns their sum.

(*iii*) Function prototype for a function that takes a reference parameter.

(*iv*) Declaration of a `Class` with a Member `Template` Function.

(*v*) Declaration for an array of pointers to integers (array holding integer pointers).

(*vi*) The prototype of a `destructor` function.

P.T.O

(*b*) Find the errors, if any in the following C++ code and rewrite the correct code :

(*i*)
```
#include <iostream>
using namespace std;
int main().
{   int a=5;
    cout<<b<<endl;
    return 0;
}
```

(*ii*)
```
#include<iostream>
using namespace std;
int fun (int x, int y)
{   return x + y;
}
double fun(int x,int y)
{     return x * y;
}
int main()
{   cout<<fun(5,10);
    return 0;
}
```

(*iii*)
```
#include <iostream>
using namespace std;
void recursiveFunction()
{   recursiveFunction();
}
int main()
{   recursiveFunction();
    return 0;
}
```

(*iv*)
```
#include <iostream>
using namespace std;
int sum(int a, int b)
{ cout<<"Adding "<<a<<" and"<<b<<endl;
}
int main()
{   int result=sum(5,10);
    cout<<"Result: "<<result<<endl;
    return 0;
}
```

(v)
```
#include <iostream>
using namespace std;
class Car
{ public:
     Car(int speed, string color)
     {   this->speed=speed;
         this->color=color;
     }
     void display()
     {    cout<<"Speed: "<<speed<<
          ", Color:"<<color<<endl;
     }
  private:
     int speed;
     string color;
};
int main()
{   Car car;
    car.display();

   return 0;
}
```

(c) Find the output of the following C++ code after the execution by giving suitable reasons : 2

```
#include <iostream>
using namespace std;
int main()
{   int x=10,y=5,z=0;
    z= x > y ? x : y > 0 ? y : 0;
    cout<<"Result: "<<z<<endl;
    return 0;
}
```

(*d*) What will be the output of this program ? Describe how the recursive calls handl both even and odd exponents.

```
#include <iostream>
using namespace std;
int power(int x, int y)
{   cout<<"power("<<x<<", "<<y<<")"<<endl;
    if (y == 0) return 1;
    int temp = power(x, y / 2);
    if (y%2 == 0)
        return temp*temp;
    else
        return x*temp*temp;
}
int main()
{   cout<<"Result: "<<power(2,5)<<endl;
    return 0;
}
```

(*e*) Write a function to search a given element in a set of N numbers using Binary Search. 4

(*f*) How can you calculate the number of elements between two pointers pointing to different elements within the same array ? Write a C++ program to demonstrate this. 4

Section B

2. (*a*) Suppose p, q, and r are int variables and p = 8, q = 4. What value is assigned to p, q, and r after each statement is executed? 3

```
p = (q--) * 2;
r = 5 * p + (++q);
```

(*b*) Write a function `isPrime(int n)` that checks if a given number *n* is prime. W a `main` function that reads an integer `N` from the user and uses `isPrime` to pr whether the number is prime or not.

(*c*) What is encapsulation in OOP, and why is it important ? Describe how encapsulat helps in maintaining and securing code.

3. (*a*) Find the errors in the following program and give the correct statements :

(*i*)
```
#include <iostream>
using namespace std;
int main()
{   int i = 1, sum = 0;
    while (i <= 10)
      sum += i;
      i++;
 cout << "Sum: " << sum << endl;
 return 0;
}
```

(*ii*)
```
#include <iostream>
using namespace std;
int main()
{   char c = 100;
    int num = "Hello World";
    cout << "Value: " << c << endl;
    cout << "Value: " << num << endl;
    return 0;
}
```

(*iii*)
```
#include<iostream>
using namespace std;
class des
{ public:
    des(int x)
    { cout<<"Constructor called with
      value:"<<x<<endl;
    }
    ~des(int x)
     { cout<<"hi";}

};
int main()
{   des obj(10);
    return 0;
}
```

(*b*) Write a program in C++ having the following properties : 9

- Define a `class` having two data variables of integer type in `private` scope. `Array of objects` should be declared in `main` function.
- Input the data variables using a member function named as `void get_data()` in `public` scope.
- Increment the values of data variables by one using a member function named `void update_data()` in `public` scope having array of object as an argument.
- For displaying the updated value of data variables, create a member function named as `void showdata()` in `public` scope.

4. Create a `class` called `LibraryBook` to track books in a library. The `class` should meet the following specifications : 2+4+3+6=15

(*i*) Data Members :

- `bookID` (an integer to uniquely identify each book).
- `title` (a string representing the title of the book).
- `copies` (an integer to store the number of copies of the book available).
- A `static` integer data member `totalBooks` to keep track of the total number of books created.

(*ii*) Constructor Requirements :

- Define a parameterized constructor to initialize `bookID` and `title` with user-specified values. If the number of `copies` is unknown, it should be set to a default of 0 using the constructors default arguments.
- Define a copy constructor that initializes a new book object as a copy of an existing `LibraryBook` object. This should increment `totalBooks` but not modify copies for the new object.

(*iii*) Member Functions :

- `addCopies(int n)`: This function should add n copies to the copies cou
- `displayBookInfo()`: This function should display the `bookID, titl` and `copies` for each book.
- A static function `getTotalBooks()` that returns the total count `LibraryBook` objects created.

(*iv*) Program Requirements :

- Create a few `LibraryBook` objects to test the class functionality, using t parameterized constructor and the copy constructor.
- Display the information of each book using `displayBookInfo()`.
- Display the total number of `LibraryBook` objects created using `getTotalBooks()`.

5. (*a*) Write a C++ program that defines a `template` function `max` which accepts tv parameters of any data type and returns the maximum of the two values. Test the functi with `int, float,` and `char` types.

(*b*) Design a C++ program to demonstrate multilevel inheritance using the following classe `Class Company` (Base Class) with attributes: `company_name` and `locatio`

`Class Team` (Intermediate Class) : Inherits from `Company` and adds attribute `team_name` and `project`.

Class `Developer` (Derived Class) : Inherits from `Team` and adds attribute `developer_name` and `programming_language`.

Use a parameterized constructors to initialize all the class attributes. Create an obj of `Developer`, and display details from all the classes: `Company, Team,` a `Developer`.

6. *(a)* 6

```
#include <iostream>
using namespace std;
class A
{ public:
    virtual void print()
    {   cout << "A::print\n";
    }
};
class B : public A
{ public:
    void print()
    {   cout << "B::print\n";
    }
};
class C : public B
{ public:
    void print()
    {   cout << "C::print\n";
    }
};
int main()
{   A *a;
    C c;
    a = &c;
    a->print();
    return 0;
}
```

(i) What will be output of the following program after the execution ? Specify appropriate reasons.

(ii) If the `virtual` keyword was removed from the `print` function in the `class A`, what would be the output of the program ? Explain why.

(*iii*) What role does polymorphism play in this code, and how does it improve flexibility in an object-oriented program ?

(*b*) Write a C++ program that performs basic arithmetic operations (addition, subtraction, multiplication, division). Include exception handling to : Handle division by zero. Handle invalid input for non-numeric data. Handle invalid operator input. The program should allow the user to input two numbers and choose an operator, and then display the result or the appropriate error message. 9

7. (*a*) Find the output considering the following scenarios. Give appropriate reasons : 6

(*i*) Suppose `numPtr` is a pointer to an integer and the size of an `int` is 4 bytes. Given, `numPtr = 5000`. What will be the new value of `numPtr`, if you perform the following pointer arithmetic:

```
numPtr = numPtr + 3 ?
```

(*ii*) Given the array `int arr[ ] = {1, 2, 3, 4, 5}` and `ptr` is a pointer initialized to point to the first element of `arr`, what will the expression `*(ptr + 4)` return ?

(*iii*)
```
#include <iostream>
using namespace std;
void square(int *snum)
{   cout << "square of 10 is ";
    *snum *= *snum;
}
int main()
{   int num = 10;
    square(&num);
    cout << num;
    return 0;
}
```

(*b*) Write a C++ program that opens a file named `"sample.txt"` in read mode and displays only the first 10 lines. If the file has fewer than 10 lines, it displays all available lines. 9