

Replace ellipses ... with proper code to override method display\_speed() in class Bike and Car. Use runtime polymorphism to display the contents of objects of classes Vehicle, Car, and Bike.

- (b) Differentiate between call by value and call by reference using an appropriate example. (4)
7. (a) Write a program in C++ to read a file info.txt and display the number of words in the file. (5)
- (b) Consider the example given below. The weather department records the rainfall (in cms) for 4 weeks in a 2D matrix. Write a code to calculate the average rainfall for each week. (5)

	Mon	Tue	Wed	Thur	Fri	Sat	Sun
Week 1	2.2	3	5	1	0.9	1.1	1.2
Week 2	1.6	2.1	1.7	2.6	5.1	0.8	1.2
Week 3	1.7	1.9	1.5	3.7	3.9	1.5	4.1
Week 4	0.6	0.5	1.3	3.3	1.5	1.9	2.5

(1000)

[This question paper contains 12 printed pages.]

Your Roll No.....

Sr. No. of Question Paper : 4361

G

Unique Paper Code : 32341101

Name of the Paper : Programming Fundamentals  
Using C++

Name of the Course : B.Sc. (H) Computer Science

Semester : I

Duration : 3 Hours

Maximum Marks : 75

**Instructions for Candidates**

1. Write your Roll No. on the top immediately on receipt of this question paper.
2. Question Number 1 is compulsory in Section A.
3. Attempt any 4 questions from Section B.
4. Parts of a question should be attempted together.

**Section A**

1. (a) Give the C++ declarations for the following :  
(5)

P.T.O.

- (i) Array of type double that holds 10 values.
  - (ii) Opening a file val.txt in write mode.
  - (iii) Pointer to character variable x initialised to 'A'.
  - (iv) Function prototype that accepts variables x and y where x is of type double and y is an array of integer.
  - (v) Declare a variable t of type integer, initialised to value 13 and use suitable manipulators to display the value in octal format.
- (b) Identify all the syntax errors (if any) in the given code and rewrite the correct code:

(i) `for (i=0, i<10, i++)` (2)

```

{
    cout<<"Square is: ", i*2, "\n";
}

```

(ii) `void area(int length=8, int breadth)` (2)

```

{
    return length*breadth;
}
cout<<"Area of the rectangle is:
"<<area (8, 20);

```

```

class Vehicle {
    int max_speed;
public:
    Vehicle() {
        max_speed = 10;
    }
    virtual void display_speed() {
        cout<<"Maximum speed of vehicle is: "<<max_speed;
    };
    class Bike: public Vehicle {
        public:
        Bike () {
            max_speed = 60;
        }
        ....
    };
    class Car: public Vehicle {
        public:
        Car() {
            max_speed =100;
        }
        ....
    };

```

5. (a) Write a function in C++ that takes a given number  $n$  and displays the square root if the number is positive otherwise throw an exception with a message – “Negative number found”. (4)

(b) Explain copy constructor with the help of an example. (3)

(c) Give the output of the following C++ code snippet. Assume all the header files have been included (3)

```
int foo(float a) {
    return a/2;
}
int foo(int a) {
    return a*2;
}
void main() {
    float a = 3.5;
    cout<<foo(a);
}
```

6. (a) Create the following hierarchy in C++ and implement the functions as mentioned below : (6)

(iii) class AA { (3)

```
public:
    void print(int x);
    int AA(int);
private
    int size;
    int max;
}
```

(c) Give the output of the following. Assume all necessary header files have been included. (2)

```
(i) void main () {
    int a = 10;
    int b = (a%2)? 10: 100;
    cout << b;
}
```

(ii) void main() { (3)

```
int x = -1;
try {
    cout << "Inside try n";
    if (x < 0)
    {
        throw x;
        cout << "After throw n";
    }
}
```

```

catch (int x) {
    cout << "Exception Caught n";
}
cout << "After catch n";
}

```

(iii) char val = 'a'; (3)

```

switch(val) {
    case 'a':
        cout<<"case a executed ";
    case 'b':
        cout<<"case b executed ";
        break;
    default:
        cout<<"Default block executed";
}

```

(iv) void passing(int &a, int b){ (3)

```

    a = a+b;
    b= b-a; }
void main() {
    int a = 7, b = 6;
    passing(a,b);
    cout<<"a = "<<a;
    cout<<"b = "<<b;
}

```

(b) Consider the following C++ class: (8)

```

class location
{
    int latitude, longitude;
}

```

where latitude and longitude are the geographic coordinates of a location. For the above class, provide the definition of the following member functions :

(i) Parameterised Constructor

(ii) Overload the "-" Operator to subtract two locations.

(iii) Display Function.

Define the main function. Use operator overloading to create two locations and store the difference between these two locations and display the result. For example, two locations, 10 latitude, 15 longitude, and 6 latitude, 9 longitude will give the result 4 latitude, 6 longitude when subtracted.

- (b) Explain the difference between function overloading and function overriding with suitable example. (4)
3. (a) Give the output with an explanation for the following. Assume all necessary header files have been included. (5)
- ```
string str1 ("Hello");
string str2 ("HELLO");
cout<<str1.compare(str2)<<"\n";
cout<<str1.substr(3,10)<<"\n";
cout<<str2.find("l")<<"\n" ;
cout<<str2.replace(1, 2, "el")<<"\n";
cout<<str1.insert(4, str2);
```
- (b) Write a function in C++ using command line arguments that adds all the numbers passed as command line arguments and display the sum. (5)
4. (a) Explain pure virtual functions by giving an example. (2)

- (v) class base { (5)
- ```
public:
    base() {
        cout<<"In base"<<"\n";
    }
    ~base() {
        cout<<"Base destructor\n";
    }
};
class derived: public base {
public:
    derived () {
        cout<<"In derived"<<"\n";
    }
    ~derived() {
        cout<<"Derived destructor\n";
    }
};
void main() {
    derived d;
    cout<<fixed<<setprecision(2)<<13.4546<<" \n";
}
```

- (d) Give the output with justification. Assume memory address of val to be 1024, memory address of ptr1 to be 1078 and memory address of ptr2 to be 1154 : (3)

```
#include <iostream>
int main()
{
    int val =5;
    int *ptr1 = &val;
    int **ptr2 = &ptr1;
    cout<<"*ptr1 = "<<*ptr1<<"\n";
    cout<<"*ptr2 = "<<*ptr2<<"\n";
    cout<<"**ptr2 = "<<**ptr2<<"\n";
    return 0;
}
```

- (e) Convert the following for-loop to equivalent do-while loop : (4)

```
for(int i =0; i<100; i= i*3)
{
    for (int j = 10; j<100; j++)
        cout<<i*j;
    cout<<"\n";
}
```

### Section B

2. (a) Define a class in C++ weightConverter with the following features : (6)

1kg = 2.2 pounds

- (i) Declare the Data Members: weightKG, weightPounds of type double
- (ii) Define the parameterised constructor to initialize weightKG.
- (iii) Define the member functions with appropriate prototypes for the following task:
  1. Converting WeightKG to WeightPounds
  2. Write an inline function averageWeight() that accepts one weight in Kilograms and another in Pounds and returns the average weight of both (in the same units)
- (iv) Define the main function to initialize the object of weightConverter and call the appropriate member function to convert weight of 50 kgs in pounds.