***DOMINIC KIPLANGAT***

***BBIT/MG/1043/09/20***

***COMP 215/BBIT 215: OBJECT ORIENTED PROGRAMMING WITH C++***

***CAT 1/ASSIGNMENT***

1. **Write a program in C++ that passes two arrays A and B to a function sum() such that the function adds the corresponding elements of both the arrays and display them**

#include <iostream>

using namespace std;

#include <algorithm>

int main()

{

int A[3] = {1, 2, 3};

int B[3] = {1, 2, 3};

transform(A, &A[0] + 3, B, A, plus<int>());

for(int i = 0; i < 3; ++i)

{

cout << A[i] <<endl;

}

return 0;

}

1. **Design a C ++ program that will accept user data input and store them in an array**

#include <iostream>

using namespace std;

int main()

{

int numbers[5];

cout << "Enter 5 numbers: " << endl;

for (int i = 0; i < 5; ++i)

{

cin >> numbers[i];

}

cout << "The numbers are: ";

for (int n = 0; n < 5; ++n)

cout << numbers[n] << " ";

return 0;

}

1. **With an appropriate C++ program code briefly explain the concept of function overloading**

**Function Overloading** is defined as the process of having two or more function with the same name, but different in parameters. In function overloading, the function is redefined by using either different types of arguments or a different number of arguments. It is only through these differences compiler can differentiate between the functions.

#include <iostream>

using namespace std;

void print(int i) {

cout << " Here is int " << i << endl;

}

void print(double f)

{

cout << " Here is float " << f << endl;

}

void print(char const \*c)

{

cout << " Here is char\* " << c << endl;

}

int main()

{

print(10);

print(10.10);

print("ten");

return 0;

}

1. **Write a simple program in C++ to find the factorial of a number using the recursive function**

#include<iostream>

using namespace std;

int factorial(int n);

int main()

{

int n;

cout<<"enter a positive integer:";

cin>>n;

cout<< "Factorial of"<< n<<"="<<factorial(n);

return 0;

}

int factorial(int n)

{

if (n>1)

return n\*factorial(n-1);

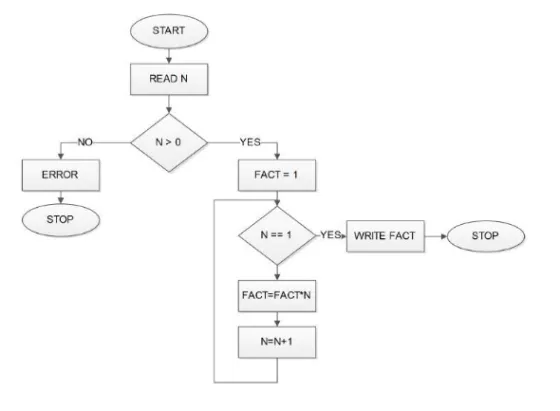
else

return 1;

}

1. **Construct a well labeled flowchart to depict the logic of program execution in question() above.**

FLOWCHART for the above



1. **Write a program in C ++ to demonstrate how a programmer can achieve inheritance access control.**

#include<iostream>

using namespace std;

class Vehicle{

public:

Vehicle()

{

cout << "This is a Vehicle\n";

}

};

class fourWheeler: public Vehicle

{

public:

fourWheeler()

{

cout << "Objects with 4 wheels are vehicles\n";

}

};

class Car: public fourWheeler

{

public:

Car()

{

cout << "Car has 4 Wheels\n";

}

};

int main()

{

// Creating object of sub class will

// invoke the constructor of base classes.

Car obj;

return 0;

}

1. **Write a program in C++ to demonstrate how to get the length of a string using Strlen() function**

#include <iostream>

#include <cstring>

using namespace std;

int main()

{

// initialize C-string

char song[] = "We Will Rock You!";

// print the length of the song string

cout << strlen(song);

return 0;

}

1. **Design a program in C++ to demonstrate the implementation of WHILE….LOOP control structure.**

#include<iostream>

using namespace std;

int main ()

{

int i=1;

while (i<5)

{

cout<< i <<endl;

i++;

}

return 0;

}

1. **Construct a well labeled flowchart to depict the logic of program in question (C) above**

Question on function overloading

